# EXHIBIT 2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

JUNIPER NETWORKS, INC.,

Petitioner,

v.

SWARM TECHNOLOGY LLC,

Patent Owner.

Case IPR2021-01317

U.S. Patent No. 9,146,777

**CORRECTED PETITION FOR**
***INTER PARTES* REVIEW OF U.S. PATENT NO. 9,146,777**

# TABLE OF CONTENTS

## EXHIBIT LIST[1]

| Exhibit | Description |
|---|---|
| 1001 | U.S. Patent No. 9,146,777 ("the '777 Patent," "the '777") |
| 1002 | Prosecution History for U.S. Patent No. 9,146,777 ("'777 History") |
| 1003 | Declaration of Dr. Jon B. Weissman ("Weissman") |
| 1004 | Curriculum vitae of Dr. Weissman |
| 1005 | U.S. Patent No. 6,006,249 ("Leong") |
| 1006 | "The Ethernet – A Local Area Network, Data Link Layer and Physical Layer Specifications," Version 1.0, September 30, 1980. (available at https://gordonbell.azurewebsites.net/Digital/Ethernet%20Blue%20Book.pdf) (last accessed April 13, 2021) ("Ethernet," "the Ethernet standard") |
| 1007 | U.S. Application Publication No. 2007/0074207 ("Bates") |
| 1008 | "Comparison of Microsoft Windows versions," PDF obtained from https://en.wikipedia.org/wiki/Comparison_of_Microsoft_Windows_versions (last accessed March 18, 2021) |
| 1009 | "Official List Announcement: RC5-56 completion," https://blogs.distributed.net/1997/10/22/00/00/bovine/ (last accessed March 18, 2021) |
| 1010 | D. Anderson *et al*. "High-Performance Task Distribution for Volunteer Computing," First IEEE International Conference on e-Science and Grid Technologies, pp. 5-8 December 2005, Melbourne (available at http://boinc.berkeley.edu/boinc_papers/server_perf/server_perf.pdf) (last accessed March 18, 2021) |
| 1011 | Y. Shoham, "Agent-oriented programming," Artificial Intelligence, vol. 60, pp. 51-92 (1993) ("Shoham") |
| 1012 | M. Burgin, G. Dodig-Crnkovic, "A Systematic Approach to Artificial Agents," In Computer Science (2009) (available at http://arxiv.org/abs/0902.3513) ("Burgin") |
| 1013 | U.S. Patent No. 9,189,281 ("Chen") |

[1] Aside from exhibits that are patents or declarations, exhibit citations are to the footer located on the bottom-right of each page.

| Exhibit | Description |
|---|---|
| 1014 | James F. Kurose & Keith W. Ross, Computer Networking, A Top-Down Approach, pp. 469-473 (4th ed. 2008) ("Kurose") |
| 1015 | "Success of Motions to Stay Rising, But Why?" (https://www.sternek-essler.com/sites/default/files/2020-03/success_of_motions_to_stay_rising_but_why.pdf, last accessed Apr. 13, 2021) |
| 1016 | Case Docket for *Juniper Networks Inc. v. Swarm Tech. LLC*, 3:20-cv-03137 (last accessed July 23, 2021) |

## CLAIM LISTING

| **Independent Claim 1** | |
|---|---|
| 1.Pre | An apparatus for parallel processing of a large computing requirement, the apparatus comprising: |
| 1.1 | a central processing unit ("CPU"); |
| 1.2 | a task pool in electronic communication with the CPU; and |
| 1.3 | a first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising a first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process; |
| 1.4 | wherein the CPU populates the task pool by dividing the requirement into one or more threads and placing the threads in the task pool, each thread comprising one or more tasks, and the matching task being one of the tasks; |
| 1.5 | wherein each task comprises a descriptor, the descriptor containing at least: |
| 1.6 | a function to be executed; and |
| 1.7 | a memory location of data upon which the function is to be executed; |
| 1.8 | wherein the first agent is a data frame comprising: a source address, a destination address and a payload; |
| 1.9 | wherein the first agent retrieves the matching task by: |
| 1.10 | being dispatched by the first solidarity cell to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the first solidarity cell is configured to perform; |
| 1.11 | searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform; and |
| 1.12 | returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address, and the payload comprises the descriptor of the matching task. |

| **Dependent Claims 2-14** | |
|---|---|
| 2 | The apparatus of claim 1 wherein the task pool notifies the CPU when the tasks of a thread are completed. |
| 3 | The apparatus of claim 1 wherein the tasks each comprise a task type selected from a set of task types, and wherein the first solidarity cell is configured to perform tasks of one or more of the task types. |
| 4 | The apparatus of claim 3 wherein the matching task is a task that is ready to be processed and has a task type that the first solidarity cell can perform. |
| 5 | The apparatus of claim 4 wherein the first agent retrieves the matching task by: searching the task pool for a task that is ready to be processed and has a task type that the first solidarity cell can perform; and identifying the matching task. |
| 6 | The apparatus of claim 1 wherein the matching task is a task that is ready to be processed and the function of the matching task can be performed by the first solidarity cell. |
| 7 | The apparatus of claim 6 wherein the first agent retrieves the matching task by searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform, and identifying the matching task. |
| 8 | The apparatus of claim 1 wherein the descriptor further contains a memory location where processed data is to be stored. |
| 9 | The apparatus of claim 1 wherein the descriptor is a data structure and the task contains a reference to the memory location of the descriptor. |
| 10 | The apparatus of claim 1 wherein the task pool occupies a region of physical memory. |
| 11 | The apparatus of claim 10 wherein the task pool is disposed in a hardware block dedicated to the task pool. |
| 12 | The apparatus of claim 1 further comprising a second solidarity cell comprising a second agent that proactively retrieves matching tasks from the task pool for the second solidarity cell to process, wherein the matching task for each solidarity cell is a task in the task pool that is ready to be processed and can be performed by the solidarity cell. |
| 13 | The apparatus of claim 12 wherein each solidarity cell sends its agent to the task pool when the solidarity cell does not have a matching task to process. |

| **Independent Claim 14** | |
|---|---|
| 14.Pre | An apparatus for parallel processing of a large computing requirement, the apparatus comprising: |
| 14.1 | a central processing unit ("CPU"); |
| 14.2 | a task pool in electronic communication with the CPU; |
| 14.3 | first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising a first agent configured to proactively retrieve, from the task pool, a matching task for the solidarity cell to process; and |
| 14.4 | a second solidarity cell comprising a second agent that proactively retrieves matching tasks from the task pool for the second solidarity cell to process, wherein the matching task for each solidarity cell is a task in the task pool that is ready to be processed and can be performed by the solidarity cell; |
| 14.5 | wherein each solidarity cell sends its agent to the task pool when the solidarity cell does not have a matching task to process; and |
| 14.6 | wherein each agent comprises a source address, a destination address, and a payload, and wherein each agent retrieves a matching task by: |
| 14.7 | being dispatched by its solidarity cell to the task pool, during which the source address is its solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the agent's solidarity cell is configured to perform; |
| 14.8 | searching the task pool for a task that is ready to be processed and has a function that the agent's solidarity cell can perform; and |
| 14.9 | returning to its solidarity cell, during which the source address is the task pool's address, the destination address is the agent's solidarity cell's address, and the payload comprises a descriptor of the matching task. |

## MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(a)(1)

- **§ 42.8(b)(1):** Juniper Networks, Inc. is the real party-in-interest.
- **§ 42.8(b)(2):** The ’777 Patent is at issue in the following cases that may affect, or be affected by, a decision in this proceeding: *Juniper Networks, Inc. v. Swarm Technology LLC*, No. 3:20-cv-03137-JD (N.D. Cal.).
- **§§ 42.8(b)(3), (4):** A Power of Attorney accompanies this Petition. Juniper consents to email service at adam.allgood.ipr@fischllp.com.

| Lead Counsel | Back-up Counsel |
|---|---|
| Adam A. Allgood (Reg. No. 67,306)<br>*adam.allgood.ipr@fischllp.com*<br>FISCH SIGLER LLP<br>5301 Wisconsin Avenue NW<br>Fourth Floor<br>Washington, DC 20015<br>Phone: 202.362.3536 | Kyle K. Tsui (Reg. No. 62,602)<br>*kyle.tsui@fischllp.com*<br>FISCH SIGLER LLP<br>5301 Wisconsin Avenue NW<br>Fourth Floor<br>Washington, DC 20015<br>Phone: 202.362.3527 |

* * * *

## USE OF EMPHASIS IN QUOTATIONS

Quotations and exhibit emphases have been added unless otherwise indicated.

## APPLICABLE STATUTES

References to 35 U.S.C. §§ 102 and 103 are to the pre-AIA versions applicable to the ’777 Patent.

# I. INTRODUCTION

U.S. Patent No. 9,146,777 purports to have contributed autonomous co-processors that "proactively seek tasks to perform" to the state of the art. The '777 Patent alleges its design avoids the overhead from managing processing tasks and reduces processor idle time. But what the '777 Patent claims wasn't new or non-obvious to a person of ordinary skill in the art.

Like the '777 Patent, Leong describes a system with independent processing units and no central control. Leong divides jobs into tasks for execution. Leong's processing units independently and proactively execute tasks obtained from a central bulletin board containing multiple tasks. And Leong's processing units communicate with each other using standard network protocols such as the Ethernet standard. But the Examiner wasn't aware of Leong and its teachings.

The Examiner only allowed the claims after finding the cited references didn't teach an agent retrieving matching tasks. But as this Petition demonstrates, this would have been obvious to a person of ordinary skill in the art when considering Leong's system with multiple processing units.

Nothing described or claimed in the '777 Patent is new or non-obvious. Instituting a trial here for Grounds 1-3, all of which are based on Leong, would be an effective alternative to litigation. Juniper Networks, Inc. thus requests *inter partes* review of claims 1-14 of the '777 Patent.

## II. STANDING AND IDENTIFICATION OF CHALLENGE UNDER §§42.104(A)-(B)

Juniper certifies the '777 Patent is available for IPR. Juniper isn't barred or estopped from requesting IPR on the asserted grounds.

Juniper asks the Board find claims 1-14 ("Challenged Claims") unpatentable based on the following patents and printed publications in view of the general knowledge of a person having ordinary skill in the art ("POSITA") prior to the purported invention:

| Short Name | Publication/Issue Date | Prior Art Status |
|---|---|---|
| Leong | **12/21/1999** | §102(b) |
| Ethernet specification | **9/30/1980** | §102(b) |
| Bates | **3/29/2007** | §102(b) |

| | Proposed §103(a) Challenges |
|---|---|
| 1 | Leong renders obvious claims 1-14. |
| 2 | Leong and Ethernet render obvious claims 1-14. |
| 3 | Leong and Bates, or Leong, Ethernet, and Bates render obvious claim 2. |

Nothing in the file history suggests the Examiner knew of Leong, and the Examiner didn't cite the Ethernet specification. And although the Examiner applied Bates during prosecution, it wasn't considered in combination with Leong and/or Leong and the Ethernet specification. Thus, Ground 1-3, and the art discussed therein, aren't cumulative of what the Examiner considered.

## III. STATE OF THE ART

By the early 2000s, physical limitations slowed the pace of semiconductor performance improvement and personal computers with multiple central processing units ("CPUs") became prolific.[2] Software developers transitioned applications and operating systems to take advantage of multiple processors. But using multiple processors to accomplish a task wasn't new.[3]

Multiprocessor systems and operating systems built to take advantage of these processors existed at least since the mid 1960s.[4] By the 1970s, mainframes and supercomputers used large numbers of processors to process transactions and solve problems in a highly parallel fashion.[5] Typically, these systems used a central managing unit coupled to multiple processing units and controlled which processing units executed particular jobs.[6] Leong's FIG. 1 depicts one of these prior art multiprocessor systems and highlights the individual processing units in yellow.

---

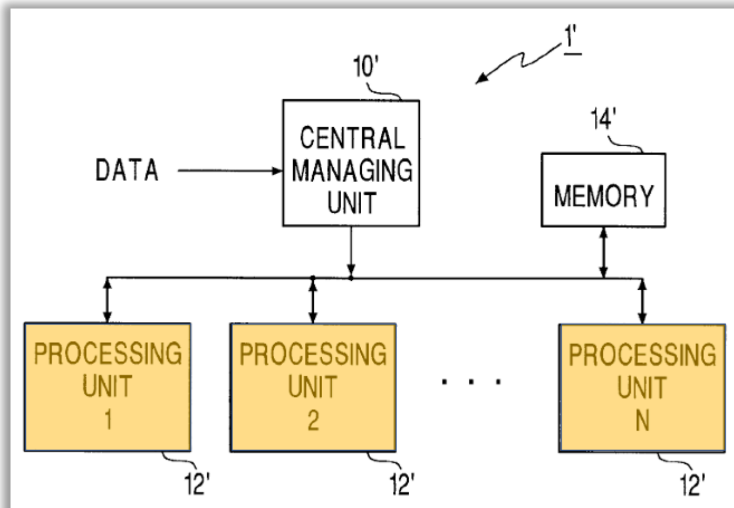
[2] Weissman, ¶¶19-20.

[3] *Id.*

[4] *Id.*, ¶20.

[5] *Id.*

[6] *Id.*; Leong, 1:20-34.




Leong FIG. 1 (Annotated)

These designs improved throughput but came with drawbacks. For one, the central managing unit imposed overhead and didn't contribute to data processing.[7] And because these were typically mainframes and supercomputers, it was difficult to expand systems by adding units because of the customized hardware and software typically used.[8] But over time, the multiprocessing concepts migrated to commodity systems, including personal computers and consumer gaming devices.

Sony released the PlayStation 3 in 2006 which used the Cell Broadband Engine Architecture ("Cell") processor.[9] The Cell processor was a multicore design

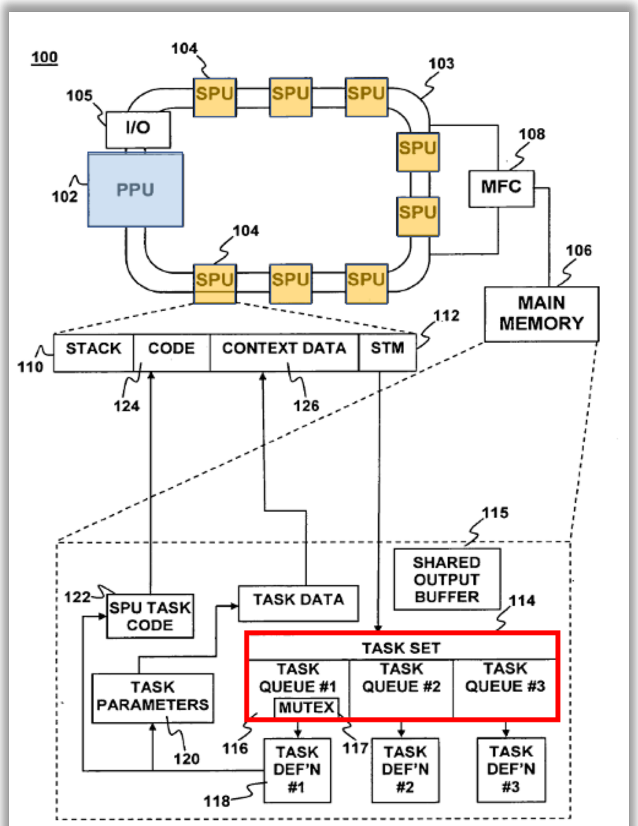
[7] Leong, 1:35-41; Weissman, ¶21.

[8] Leong, 1:42-48; Weissman, ¶21.

[9] Weissman, ¶22.

with a power processing unit ("PPU") and eight synergistic processing units ("SPUs").[10]

In FIG. 1 (below), Bates[11] depicts a Cell processor with the PPU (light blue), the SPUs (yellow), and task sets 114 (red) with task queues 116.




Bates FIG. 1 (Annotated)

[10] *Id.*; Bates, ¶7.

[11] As explained in Section IV.B, the Examiner relied on Bates to reject the claims of the '777 Patent.

The Cell's PPU "acts as a controller for the SPUs 104, which handle most of the computational workload" and hosts the operating system.[12] The PPU and the SPUs are connected to main memory and can access an area in memory with the task set. The PPU adds tasks to the task queues "but has little other involvement with the management of the task set 114."[13] Bates states this design was easier to use than prior designs and allowed software to use the multiple SPUs to execute tasks in parallel.[14]

In addition to use multiple processors in a single system, in a single processor, some distributed computing projects use the Berkeley Open Infrastructure for Network Computing ("BOINC") system so that idle volunteered computing power executes tasks. In a BOINC network, each task is independently performed by two or more computers until the network achieves a quorum of equivalent outputs.[15] BOINC clients coordinate with a server to obtain tasks and report results.[16] BOINC

---

[12] Bates, ¶33; Weissman, ¶23.

[13] Bates, ¶36; Weissman, ¶23.

[14] Bates, ¶¶15 (describing desirable improvements on the prior art), 31 (describing performance scaling with the number of SPUs), 71-72 (describing advantages over the identified prior art); Weissman, ¶¶23-24.

[15] Ex. 1010, 2; Weissman, ¶¶25-26.

[16] Ex. 1010, 1; Weissman, ¶¶25-26.

servers validate client submitted work for correctness and ensure tasks are completed.

And in computer science, software agents are well-known. A programming paradigm called agent-oriented programming ("AOP") was developed around this concept.[17] Shoham, published in 1993, characterizes an agent as "an entity that functions continuously and autonomously in an environment in which other processes take place and other agents exist."[18] These agents are "autonomous" because they don't require constant guidance or intervention.[19]

Thus, using independent processors working together in a highly independent manner to solve a large computing problem was well known and commonplace well before 2013. And the use of agents in such a computational environment was also well known before 2013.[20]

## IV. THE CHALLENGED PATENT

### A. '777 Patent Overview

The '777 Patent is titled "Parallel Processing with Solidarity Cells by Proac-

---

[17] *E.g.*, Shoham, 51; Weissman, ¶27.

[18] Shoham, 52; Weissman, ¶27.

[19] Shoham, 52; *see also* Burgin, 2; Weissman, ¶27.

[20] Weissman, ¶28.

tively Retrieving From a Task Pool a Matching Task for the Solidarity Cell to Process." The application for what issued as the '777 Patent was filed January 25, 2013.[21]

The patent discloses a parallel computing apparatus. The apparatus's CPU divides a large computing requirement into smaller tasks and posts them to a task pool.[22] The CPU doesn't control task assignment.[23] Rather, agents proactively search the task pool for matching tasks.[24] The patent asserts this avoids idle time and speeds the completion of the overall computing requirement.[25]

To process the large computing requirement, the '777 Patent describes independent computation units "capable of executing one or more tasks."[26] These units—or "solidarity cells"—may be various types of processors or standalone computers. Each system will have multiple solidarity cells. Although solidarity cells may utilize general-purpose processors or standalone computers, the patent

[21] *Id.*, ¶29.

[22] '777 Patent, 1:66-2:4, 3:15-19, 4:24-27; Weissman, ¶30.

[23] '777 Patent, 5:21-22; Weissman, ¶30.

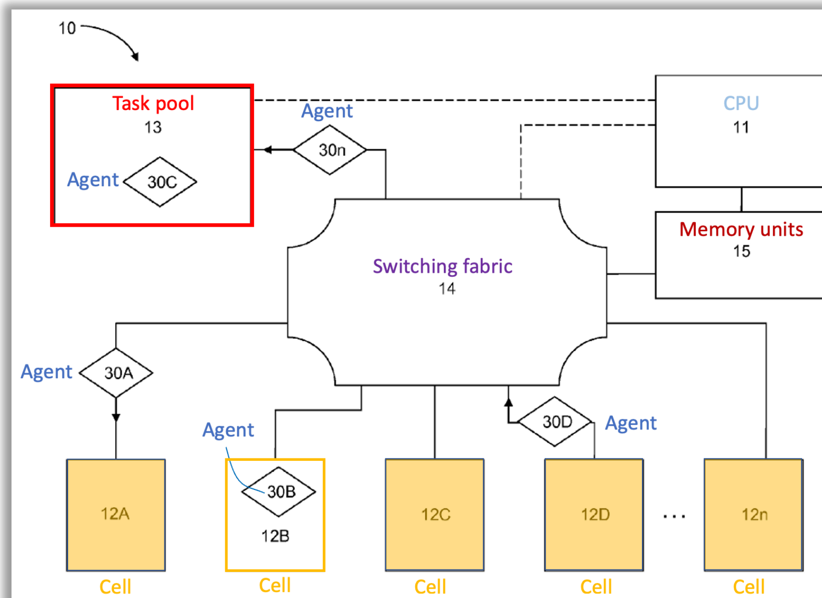
[24] '777 Patent, 5:22-27, 6:7-10; Weissman, ¶30.

[25] '777 Patent, 5:65-6:10; Weissman, ¶30.

[26] '777 Patent, 3:31-56; Weissman, ¶31.

also describes how some solidarity cells may include special-purpose processors with dedicated hardware to perform specialized tasks.[27]

The '777 Patent's FIG. 1, reproduced below shows, switching fabric 14 (purple) connects cells 12A-12*n* (yellow) so they communicate with a task pool 13 (red). With multiple cells, the system will continue task execution even if a given solidarity cell is removed from the system. And if solidarity cells are added to the system, increased performance is possible.[28]




'777 Patent FIG. 1 (Annotated)

[27] '777 Patent, 3:31-60; Weissman, ¶31.

[28] '777 Patent, 4:2-11; Weissman, ¶32.

Each cell autonomously fetches, moves, and processes data stored in memory.[29] Each cell uses agents (blue) which are "autonomous representatives of their corresponding solidarity cells."[30] Cells dispatch agents to retrieve a task any time the cell is idle or capable of performing additional processing.[31] When dispatched to the task pool, the agents include information on the types of tasks the corresponding cell can perform.[32]

### B. Prosecution History

The '777 Patent's application was filed 01/25/2013.[33] During prosecution, the Examiner rejected most of the claims as obvious over Bates either alone or in combination with U.S. Application Publication No. 2006/0026377 ("Sikdar").[34] Bates, as discussed in Section III above, describes the Cell processor. The Exam-

---

[29] '777 Patent, 4:9-11; Weissman, ¶33.

[30] '777 Patent, 5:61-62; Weissman, ¶33.

[31] '777 Patent, 2:8-13, 5:22-27, 5:61-65; Weissman, ¶¶33-35.

[32] '777 Patent, 3:57-4:11, 5:55-57; Weissman, ¶36.

[33] '777 History, 27-29 (originally-filed claims); Weissman, ¶36.

[34] '777 History, 60-79 (03/11/2015 Office Action); Weissman, ¶36.

iner turned to Sikdar for teachings as to what fields and values a TCP packet includes.[35] And the Examiner indicated claims 14 and 19 included allowable subject matter.[36]

Applicant amended the claims, including adding that the solidarity cell's agent doesn't require an instruction from the CPU to retrieve tasks from the task pool.[37] Applicant then argued Bates instead described "a typical prior art coprocessing regime in which the coprocessors are controlled by the main processor."[38]

The Examiner didn't respond to Applicant's comments.[39] Instead, an Examiner's Amendment incorporating allowable claim 14 and other dependent claims into original claim 1 was included with the Notice of Allowance.[40]

The patent issued on 9/29/2015.[41]

---

[35] '777 History, 77-78; Weissman, ¶36.

[36] '777 History, 77 (03/11/2015 Office Action); Weissman, ¶36.

[37] '777 History, 95-108, 107 (05/29/2015 Amendment); Weissman, ¶37.

[38] '777 History, 105-107 (05/29/2015 Amendment), 131-136 (Examiner's Amendment in 06/29/2015 Notice of Allowance); Weissman, ¶37.

[39] '777 History, 125-138 (06/29/2015 Notice of Allowance); Weissman, ¶38.

[40] '777 History, 138 (06/18/2015 Examiner-Initiated Interview Summary); Weissman, ¶38.

[41] '777 Patent, Cover; Weissman, ¶39.

### C. Level of Skill in the Art

At the time of the purported invention, a POSITA would have had a bachelor's degree in computer science, electrical engineering, computer engineering, or a closely related field, and one or more years of experience in the design and development of parallel and distributed processing systems. Alternatively, a POSITA would have a master's degree or similar post-graduate work in computer science, electrical engineering, computer engineering, or a closely related field, and less design and development experience.[42]

### D. Claim Construction

Only terms necessary to resolve the controversy need to be construed. For the purposes of this proceeding, Petitioner offers express constructions for "solidarity cell" and "thread." Petitioner construes all other terms as having their ordinary and customary meanings.[43]

#### 1. "solidarity cell"

Independent claims 1 and 14 recite the apparatus for parallel processing of a large computing requirement includes, in combination with other features, a first

[42] Weissman, ¶¶58-63.

[43] Consolidated Trial Practice Guide, 44; Weissman, ¶64.

"solidarity cell." Independent claim 14 and dependent claims 3-6, 12, and 13 recite a second "solidarity cell."[44]

The '777 Patent describes solidarity cells as general or special-purpose processors that act as "autonomous computer processing units."[45] Each solidarity cell is connected to the task pool by a switching fabric, and each cell is proactive. Each cell "obtains a task to perform by sending its agent to the task pool when the solidarity cell has no processing to perform."[46] A solidarity cell is an "independent computational unit capable of executing one or more tasks" and can be a microcontroller, microprocessor, or a standalone computer.[47]

The term "solidarity cell" isn't a term of art. But based on the '777 Patent, a POSITA would interpret the term to mean "an independent, autonomous proactive computer processing unit." A general-purpose processor can be such a "solidarity cell" as long as it is "independent," "autonomous," and "proactive."[48]

---

[44] Weissman, ¶65.

[45] '777 Patent, 1:60-66; Weissman, ¶66.

[46] '777 Patent, 2:4-11; Weissman, ¶66.

[47] '777 Patent, 3:31-56, 5:15-25; Weissman, ¶¶66-67.

[48] Weissman, ¶68.

### 2. "thread(s)"

Independent claim 1 recites the apparatus for parallel processing of a large computing requirement includes, in combination with other features, a CPU that "divid[es] the requirement into one or more ***threads*** and plac[es] the ***threads*** in the task pool, each ***thread*** comprising one or more **tasks**." Dependent claim 2 further recites that "the task pool notifies the CPU when the **tasks** of a ***thread*** are completed."[49]

The '777 Patent describes how "co-processors work in solidarity to complete a large computational requirement by processing ***threads*** and subtasks."[50] Further, the patent describes that "a **task** may include task ***threads***, which each contain one or more subtasks to be performed" which are transmitted to a <u>**task pool**</u>.[51] When discussing **tasks**, ***threads***, and a <u>**task pool**</u>, the patent explains:

> The <u>**task pool**</u> 13 contains one or more task ***threads*** 21. A task ***thread*** 21 represents a computational **task** that a component of the larger requirement imposed on the CPU 11. In one embodiment, the CPU 11 may initialize and then populate the <u>**task pool**</u> 13 by dividing the large requirement into concurrently executable ***threads*** 21 and places the ***threads*** 21 in the <u>**task pool 13**</u>. A ***thread*** 21 is composed of one or more

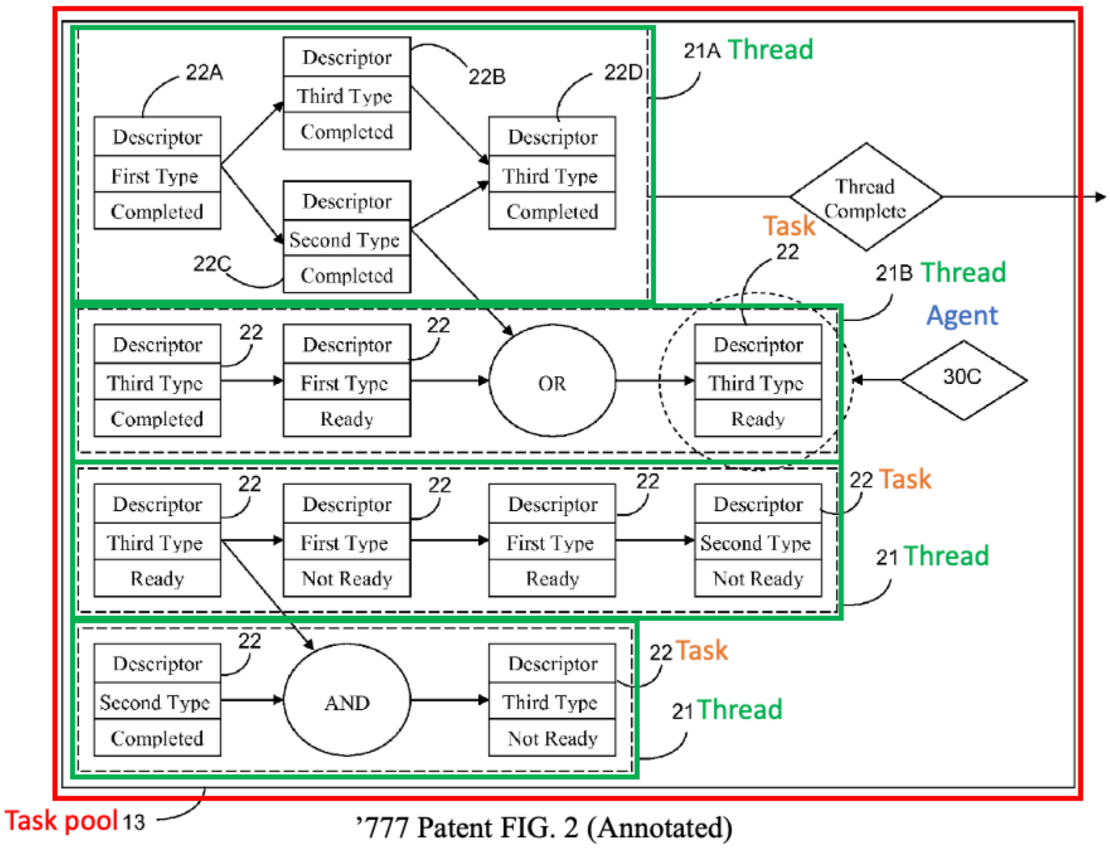
[49] *Id.*, ¶69.

[50] '777 Patent, 1:50-55; Weissman, ¶70.

[51] '777 Patent, 2:2-4; Weissman, ¶70.

**tasks** 22. A **task** 22 may have a task type and a descriptor.[52]

The tasks, threads, and the task pool are labeled in FIG. 2 below. [53]




’777 Patent FIG. 2 (Annotated)

The patent assigns other characteristics to ***threads*** consistent with the depiction in FIG. 2 where some threads appear to execute sequences (e.g., thread 21A).[54]

A ***thread*** 21 may further comprise a “recipe” describing the order in

[52] ’777 Patent, 4:21-28; Weissman, ¶70.

[53] Weissman, ¶71.

[54] *Id.*, ¶¶72-73.

> which the **tasks** 22 should be performed and any conditions that affect the order of performance. According to the recipe, the **tasks** 22 may be executed sequentially, concurrently, interdependently, or conditionally according to Boolean operations. [55]
>
> . . .
>
> ***Threads*** 21 may also be independent.[56]

As a term of art, ***threads*** has a different meaning. For example, Bates describes a thread as "a part of a program that can execute independently from other parts."[57] The '777 Patent's threads having one or more subtasks and having a "recipe" of tasks isn't the conventional usage of the term. Nor is it conventional to allocate ***threads*** to a **task pool**.[58]

A POSITA would have understood the '777 Patent's usage of ***thread*** as a shorthand version of **task thread**, which is a discrete computational task component of the large computing requirement. The patent describes a large computing

[55] '777 Patent, 4:62-5:4, FIG. 2; Weissman, ¶72.

[56] '777 Patent, 5:4-5, FIG. 2; Weissman, ¶72.

[57] Bates, ¶10; Weissman, ¶73.

[58] Weissman, ¶73.

requirement as a computation that can be broken "into a group of smaller computations for concurrent resolution."[59] And as claim 1 further clarifies, the "thread" comprises one or more subtasks to be performed.

## V. GROUND 1: CLAIMS 1-14 ARE UNPATENTABLE OVER LEONG.

Before 2013, it was well-known and ubiquitous to break up large computing requirements into smaller tasks executed using multiple processors.[60] This brought, among other benefits, increased efficiency and performance.[61] Leong discloses a system of autonomous processing units with an associated bulletin board.[62] Leong's processing units divide a computing requirement into tasks and populate those tasks to the bulletin board.[63] Leong's processing units work independently, proactively read the bulletin board, locate tasks they are capable of performing,

---

[59] '777 Patent, 1:17-24, 1:52-55, 3:49-56 (describing the plurality of solidary cells processing "the large requirement"), 4:22-27 (describing a task thread representing "a computational task that a component of the larger requirement imposed on the CPU 11"), 6:19-23 (describing the "large requirement" including tasks of various types); Weissman, ¶¶74-75.

[60] Leong, 1:20-34, FIG. 1; Bates, ¶¶7, 33; Weissman, ¶77.

[61] Weissman, ¶77.

[62] Leong, Abstract, 1:55-61, 3:9-15, 5:20-24, 6:4-13, 6:47-61, 8:34-46, FIG. 2; Weissman, ¶77.

[63] Leong, 3:9-11, 3:19-25, 4:9-24, 6:16-46, 8:65-9:4, FIGS. 3, 5, 6; Weissman, ¶77.

and execute matching tasks.[64] Leong is in the same field of endeavor as the '777 Patent and, like that patent, sought to streamline the performance of large-scale computing requirements by providing an efficient, readily scalable system that employs multiple parallel processing units.[65]

When considered by a POSITA at the time of the purported invention, Leong discloses, either expressly or inherently, or renders obvious, every element of the Challenged Claims.[66]

### A. Leong Overview

Leong was filed 8/19/1997 and issued 12/21/1999.[67] Leong describes a system of multiple processing units[68] and an electronic bulletin board.[69] Leong's processing units work independently to complete a large project by breaking up the

---

[64] Leong, 2:54-64, 3:6-18, 3:26-4:3, 4:18-35, 6:47-61, 7:34-50, FIGS. 7A, 7C; Weissman, ¶77.

[65] Leong, 1:6-10, 1:49-52, 1:55-67; *see also*,'777 Patent, 1:9-13, 1:46-55; Weissman, ¶77.

[66] Weissman, ¶¶40, 76-218, 237.

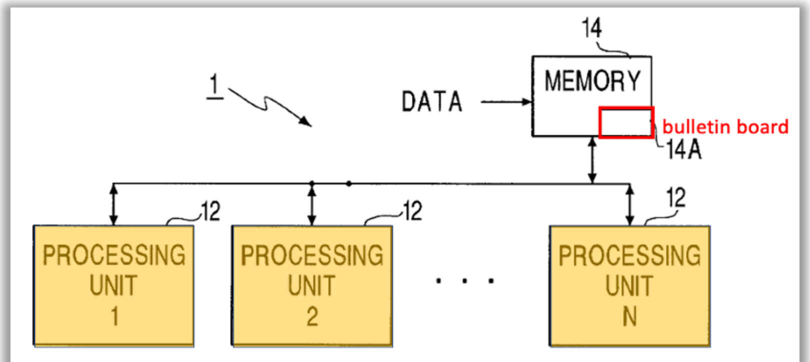
[67] Leong, Cover; Weissman, ¶¶41-42.

[68] This Petition uses the term "processing unit" consistent with Leong's FIG. 2 instead of "micro-processing units" of Leong's text.

[69] Leong, Abstract, 1:55-61, 3:9-15, 5:20-24, 6:4-13, 6:47-61, 8:34-46, FIG. 2; Weissman, ¶43.

project into smaller tasks, posting those smaller tasks to the bulletin board, and executing those smaller tasks.[70]

Leong's FIG. 2 below depicts a system with processing units (orange) and a bulletin board (red).[71]




Leong FIG. 2 (Annotated)

Each processing unit identifies executable tasks by reading the bulletin board.[72] Each task in the bulletin board includes information such as the task type, task priority, current execution status of the task, whether the task is complete, and conditions that must be met prior to task execution.[73] Using this information, each of Leong's processing units independently acquire tasks for execution.[74]

---

[70] Leong, Abstract, 2:40-46, 2:54-64, 3:9-15, 3:19-30, 4:18-35, 5:20-24, 5:62-6:3, 6:36-61, 9:1-4, FIGS. 2, 3, 6, 7A, 7C; Weissman, ¶43.

[71] Weissman, ¶44.

[72] Leong, 3:11-18; Weissman, ¶45.

[73] Leong, 3:19-4:3; Weissman, ¶45.

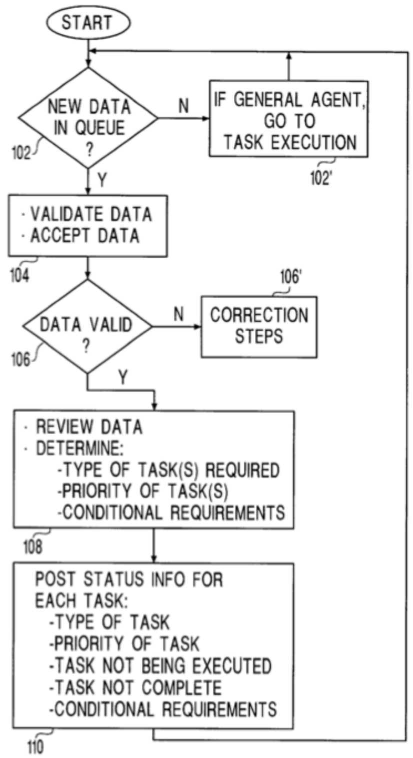
[74] Leong, 3:26-30; Weissman, ¶45.

To fill the bulletin board with tasks, one or more of Leong's processing units act as "surveying agents."[75] Leong's FIG. 6, reproduced below, shows these processing units survey (step 102) and validate (step 104) the project data set, and determine which tasks need to be performed to complete the project (step 108).[76] The surveying processing units define the required tasks' status information (step 108), then post the status information for all the required tasks to the bulletin board (step 110).[77]

[75] Leong, 4:9-24; Weissman, ¶46.

[76] Leong, 4:9-24, 6:16-39, 8:65-9:4, FIGS. 5, 6; Weissman, ¶46.

[77] Leong, 3:9-11, 3:19-25, 4:9-24, 6:36-46, 9:1-4, FIGS. 3, 6; Weissman, ¶46.




Leong FIG. 6

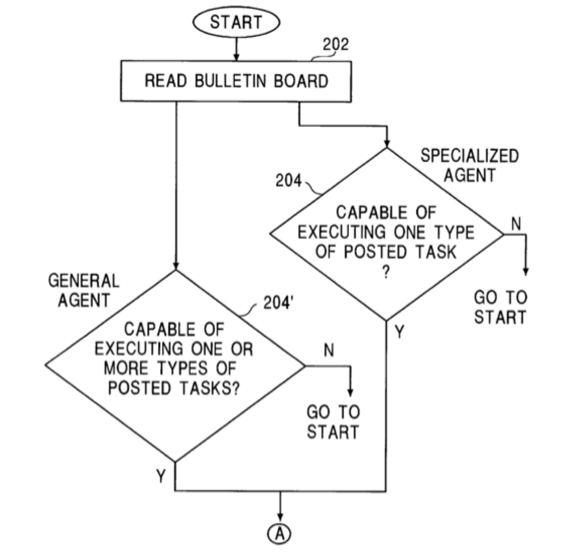
After the task status information is posted, FIG. 7A shows each processing unit performing the steps of reading the bulletin board (step 202) and determining if the unit is capable of performing any of the listed required tasks (steps 204, 204').[78] Leong's processing units may have different capabilities.[79] Some "general agents" are capable of performing more than one type of task.[80] Other processing

[78] Leong, 3:11-15, 3:26-30, 4:18-35, 6:47-61, FIG. 7A; Weissman, ¶47.

[79] Weissman, ¶47.

[80] Leong, 4:26-37; Weissman, ¶47.

units referred to as "specialized agents" are predisposed to certain types of tasks.[81] Depending on the configuration and system conditions, Leong's processing units idle themselves or execute other tasks not listed on the bulletin board until their preferred task type meets a desired threshold.[82]




Leong FIG. 7A

FIG. 7C below shows Leong's processing unit performs the step of selecting the matching task with highest priority (step 206).[83] It retrieves the data to perform the task (step 210) and changes the task's status in the bulletin board (step 212) to prevent simultaneous execution.[84] The processing unit then completes the task

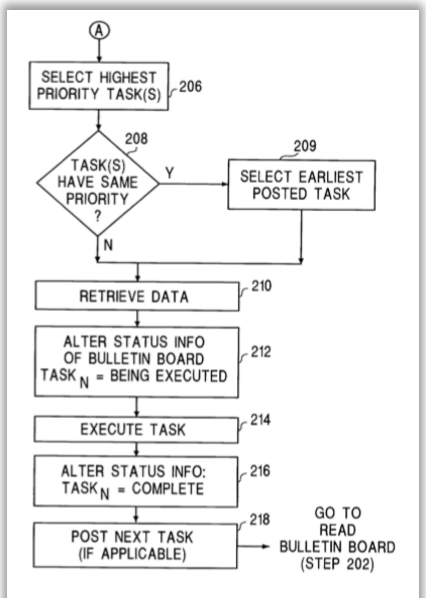
[81] Leong, 4:17-24; Weissman, ¶47.

[82] Leong, 4:30-61; Weissman, ¶47.

[83] Leong, 3:31-44, 7:34-39, FIG. 7C; Weissman, ¶48.

[84] Leong, 3:45-54, 7:39-45, FIG. 7C; Weissman, ¶48.

(step 214) and changes the task's status in the bulletin board to complete (step 216) to avoid task duplication.[85] The processing unit posts any follow-up tasks as directed in the completed task (step 218)[86] and then searches the bulletin board again for a new matching task (step 202 in FIGS. 7A, 7C).[87] Alternatively, when adding the tasks to the bulletin board, a surveying processing unit may create the necessary tasks for a task sequence and include the needed status information for a sequence to execute with the tasks.[88]




Leong FIG. 7C

[85] Leong, 3:55-61, 7:46-48, FIG. 7C; Weissman, ¶48.

[86] Leong, 3:62-4:3, 7:48-50, FIG. 7C; Weissman, ¶48.

[87] Leong, 7:51-52, FIGS. 7A, 7C; Weissman, ¶48.

[88] Leong, 3:62-4:17; Weissman, ¶48.

Leong's processing units work together, repeating this sequence until they've completed the large project's tasks.[89]

### B. Independent Claim 1

Leong alone renders obvious claim 1.[90]

#### 1. Claim 1's Preamble: "An apparatus for parallel processing of a large computing requirement, the apparatus comprising:"

To the extent the preamble limits the claim, Leong discloses all elements of the preamble.[91] Leong discloses an apparatus for parallel processing of a large computing requirement.[92] Leong's FIG. 2 depicts an apparatus 1 which includes several processing units (yellow) associated with a bulletin board (red).[93]

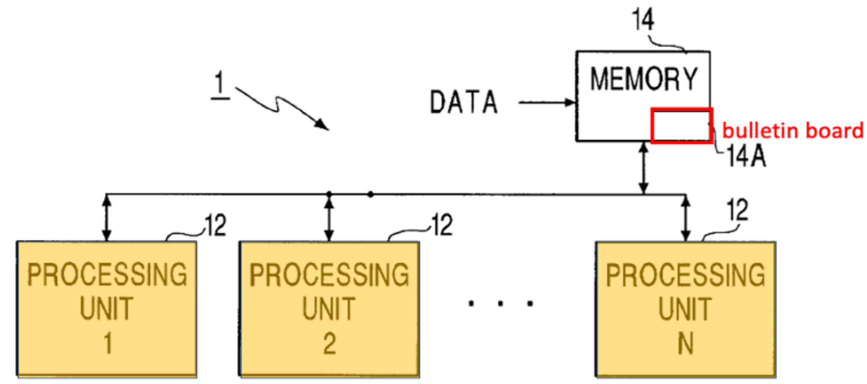
[89] Leong, 2:54-64, 3:6-18; Weissman, ¶49.

[90] Weissman, ¶79.

[91] *Id.*, ¶¶80-82.

[92] *Id.*, ¶80.

[93] *Id.*




Leong FIG. 2 (Annotated)

Leong's apparatus partitions the overall work into tasks and that those tasks can "be further broken down into smaller tasks if desired so long as the tasks achieve the required result."[94] The only requirement is the data should be partition-able into tasks.[95] Leong then executes these tasks in parallel, as processing units are available.[96] Leong's example embodiment is a check processing system for banks.[97] A POSITA would have understood such a system would have a large computing requirement.[98]

[94] Leong, 2:54-64 (describing breaking tasks into smaller tasks); Weissman, ¶81.

[95] Leong, 2:54-57; Weissman, ¶81.

[96] Leong, 6:53-55, FIG. 7A, 3:48-54; Weissman, ¶81.

[97] Leong, 8:8-23, FIGS. 8, 9; Weissman, ¶81.

[98] Weissman, ¶81.

### 2. Claim Element 1.1: "a central processing unit ("CPU");"

Leong discloses this claim element.[99] The '777 Patent describes the CPU as a conventional CPU found in a personal computer, a reduced instruction set computer, or the processors of one or more computers on a locally or remotely connected network.[100] The patent's CPU divides the large computing requirement into one or more tasks and transmits those tasks to the task pool.[101] The patent considers a CPU suitable for use if "its operating system may be programmed to recognize and communicate with the task pool … and divide computing requirements into threads."[102] The '777 Patent's disclosure contemplates any generic programmable processor as a CPU.[103]

Leong discloses multiple processing units as standalone computers coupled over a standard network.[104] Leong's units execute application software.[105] Leong's

[99] *Id.*, ¶¶83, 86, 89.

[100] '777 Patent, 2:66-3:12; Weissman, ¶84.

[101] '777 Patent, 2:1-4; *see also id.*, 4:24-27, 5:11-14; Weissman, ¶84.

[102] '777 Patent, 3:15-19; Weissman, ¶84.

[103] Weissman, ¶84.

[104] Leong, 5:20-24, 8:34-43; Weissman, ¶85.

[105] Leong, 5:64-66; Weissman, ¶85.

processing units don't need to share the same hardware and software configurations.[106]

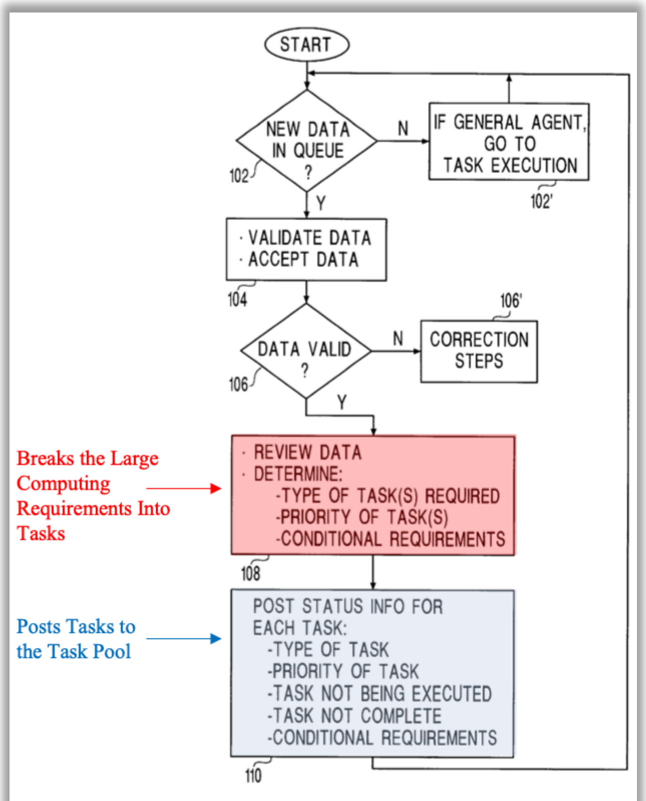
Leong's processing units "survey" the queue in memory to determine whether new data has arrived."[107] Leong characterizes this as a specialization of the processing units.[108] Leong's FIG. 6 below highlights these steps. When new data is added to the memory queue 102, the surveying agent examines the data, determines which tasks need to be performed to accomplish the overall project, defines the status information for each task (step 108, red), and posts the status information to the task bulletin board (step 110, blue).[109]

[106] Leong, 6:4-6; Weissman, ¶85.

[107] Leong, 4:10-11; Weissman, ¶87.

[108] Leong, 4:10-25; Weissman, ¶87.

[109] Leong, 4:10-17, FIG. 6; *see also*, *id.*, 6:36-46, 9:1-4, 9:42-45, FIGS. 3, 5 (at elements 100, 200); Weissman, ¶88.




Leong FIG. 6 (Annotated)

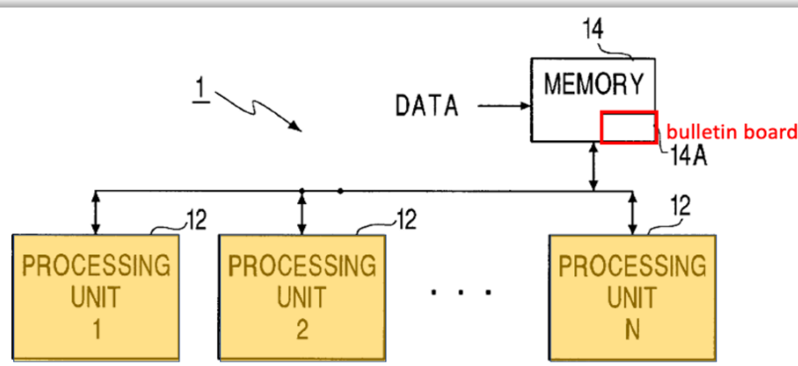
### 3. Claim Element 1.2: "a task pool in electronic communication with the CPU; and"

Leong discloses this claim element.[110] The '777 Patent describes the "task pool" as "a region of physical memory that is addressable by the CPU 11" and is

[110] Weissman, ¶¶90, 95.

"also addressable by the cells 12A . . . *n*." The task pool may be a dedicated hardware block or may be software based.[111] The patent's CPU 11 and the task pool 13 communicate either directly or through switching fabric 14.[112]

Leong's bulletin board is part of memory 14.[113] Leong's processing units, including units acting as surveying agents, read the bulletin board contents when seeking additional tasks.[114] Leong's FIG. 2 below depicts this configuration of the processing units and the bulletin board.[115]




Leong FIG. 2 (Annotated)

[111] '777 Patent, 4:12-18; Weissman, ¶91.

[112] '777 Patent, 2:56-58, 5:35-38 (describing the task pool having a network address), 6:57-67 (describing the switching fabric as a router providing connectivity between all of the system resources), FIG. 1; Weissman, ¶91.

[113] Leong, 3:9-11; Weissman, ¶92.

[114] Leong, 3:9-18; Weissman, ¶92.

[115] Weissman, ¶92.

Leong's units communicate with each other using a standard network.[116] At a minimum, Leong's FIG. 2 discloses processing units, including surveying agents read bulletin board contents.[117] Leong therefore discloses the bulletin board being in electronic communication with the processing units.[118]

And as discussed for claim element 1.1, Leong's processing units post tasks to the bulletin board.[119] Leong explains a surveying agent processing unit determines tasks to be performed on the data, defines the status information for the tasks, and posts the status information for the tasks on the bulletin board.[120]

#### 4. Claim Element 1.3: "a first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising a first agent configured to proactively retrieve, from

[116] Leong, 5:20-24, 6:10-14; Weissman, ¶93.

[117] Leong, 3:11-15; Weissman, ¶93.

[118] Weissman, ¶93.

[119] *Id.*, ¶94.

[120] Leong, 4:10-17; *see also* Leong, 3:9-11 (describing how tasks are posted to an electronic bulletin board), 3:19-25 (describing the information of each task posted to the electronic bulletin board), 4:9-24 (describing a processing unit acting as a surveying agent), 6:16-46 (describing how each of the processing units 12 can execute basic tasks including data validation and posting tasks), 8:65-9:4 (describing the check image processing embodiment), FIGS. 3, 5, 6; Weissman, ¶94.

**the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process;"**

Leong discloses this claim element.[121] This claim element recites the term "solidarity cell." As discussed in Section IV.D.1 above, "solidarity cell" means "an independent, autonomous, proactive computer processing unit."[122]

Leong's processing units are disclosures of solidarity cells.[123] Each processing unit is "self controlled using a software application running on its own operating system."[124] Leong's processing units don't require an instruction from the CPU because they "are not under the control of a central management computer or program and … operate in a substantially autonomous fashion."[125] Because of this independence, Leong's processing units don't need to have the same hardware or software configurations.[126] Leong also contemplates the processing units becoming

[121] Weissman, ¶¶96, 109.

[122] *Id.*, ¶97.

[123] *Id.*, ¶98.

[124] Leong, 2:43-47; Weissman, ¶98.

[125] Leong, 5:66-6:3, 2:43-48; Weissman, ¶98.

[126] Leong, 6:4-6; Weissman, ¶98.

disconnected or otherwise failing to execute tasks, further confirming their independence from each other.[127] Leong's processing units are therefore independent and autonomous.[128]

Leong's processing units are also proactive.[129] The processing units read the bulletin board contents and determine if they're capable of performing the tasks.[130] This is possible because the tasks indicate what type of work is involved.[131] This indication is necessary because certain processing units may be "specialized" and perform only one type of task.[132] Other agents are capable of performing multiple tasks and are "generalized agents."[133]

---

[127] Leong, 5:29-35; Weissman, ¶98.

[128] Weissman, ¶98.

[129] *Id.*, ¶99.

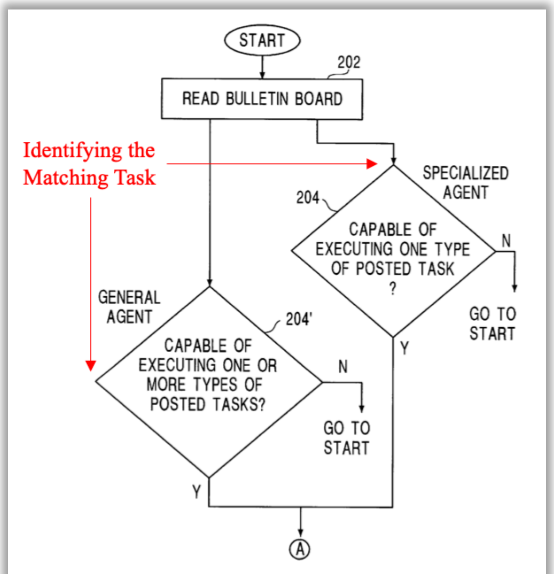
[130] Leong, 2:54-64 (describing tasks and how they may be divided into individual tasks), 3:6-18 (describing basic task processing, distribution, and control), 3:26-4:3 (describing status information for each task), 4:18-35 (describing specialized and general agents), 6:47-61 (describing FIG. 7A), 7:34-50 (describing task prioritization), FIGS. 7A, 7C; Weissman, ¶99.

[131] Leong, 3:26-30, 6:48-60, FIG. 7A; Weissman, ¶99.

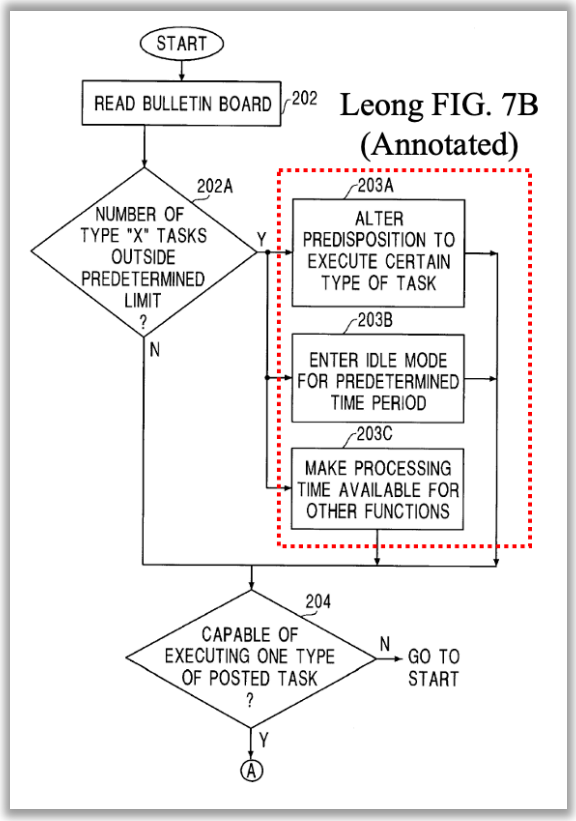
[132] Leong, 4:18-25; Weissman, ¶100.

[133] Leong, 4:26-29; Weissman, ¶100.

Further confirming the proactive nature of the processing units, Leong's FIG. 7A depicts how each processing unit independently seeks new tasks.[134] Leong's FIG. 7B shows how each processing unit can alter its predisposition to execute certain types of tasks.[135]



Leong FIG. 7A (Annotated)



Leong FIG. 7B (Annotated)

[134] Weissman, ¶101.

[135] *Id.*

Leong's processing units seek matching tasks for execution (steps 204, 204').[136] When no such tasks are available, the units' predisposition to execute certain types of tasks can change (step 203A), the units can idle for a predetermined period (step 203B), or the units' processing time is made available for other functions (step 203C).[137] Leong's processing units don't wait to receive new tasks.[138] Instead, they proactively seek out matching tasks, without any instruction from the surveying agent.[139]

As discussed for claim element 1.2, each of Leong's processing units electronically communicate with the task pool. Leong therefore discloses the recited first solidarity cell.[140]

To the extent it is argued the claims require a processor with multiple cores, such an implementation choice would have been obvious to a POSITA because the prior art confirms such multicore processors were commonplace before 2013.[141] Bates describes the Cell processor which includes a PPU and multiple SPUs.[142] A

---

[136] Leong, 6:47-61 (discussing FIG. 7A); Weissman, ¶102.

[137] Leong, 6:62-7:34 (discussing FIG. 7B), 4:50-5:19; Weissman, ¶102.

[138] Weissman, ¶102.

[139] Leong, 6:47-61; Weissman, ¶102.

[140] Weissman, ¶¶103-104.

[141] *Id.*, ¶105.

[142] Bates, ¶33, FIG. 1; Weissman, ¶105.

POSITA would have been motivated to make such an implementation choice to help reduce costs and improve performance.[143] For example, with a single processor module with multiple processing units, the amount of power needed is reduced, especially when compared to multiple standalone computers.[144]

Regarding the recited first agent being configured to proactively retrieve a matching task from the task pool for the solidary cell to process without requiring an instruction from the CPU, Leong's processing units inherently disclose the use of agents as claimed in the '777 Patent.[145] The term "agent" is often used "in such diverse ways that it has become meaningless without reference to a particular notion of agenthood."[146] At one point, the patent says agents are "autonomous representatives of their corresponding solidarity cells."[147] This is consistent with the typical usage of the "agent" term.[148] As discussed above, Leong's processing units

[143] Weissman, ¶105.

[144] *Id.*

[145] *Id.*, ¶106.

[146] Shoham, 52; Weissman, ¶106.

[147] Leong, 5:61-67; Weissman, ¶106.

[148] Shoham, 52 ("Most often, when people in AI use the term 'agent', they refer to an entity that functions continuously and autonomously in an environment in which other processes take place and other agents exist."); Weissman, ¶106.

each proactively retrieve matching tasks from the bulletin board and don't require instruction from the CPU.[149]

The '777 Patent agents "may be considered a data frame in the networking sense."[150] Although the patent also characterizes agents as "autonomous represent-atives of their corresponding solidarity cells" that "may be dispatched by their cor-responding cells … to retrieve a task," a POSITA would have understood this de-scription applies to a standard data frame.[151] A data frame is dispatched, or sent, from a source to a destination.[152] And a data frame is independent and autonomous from Leong's processing units.[153]

Leong's processing units use a standard Ethernet network which requires data frames.[154] A POSITA would have therefore understood Leong's processing

[149] Weissman, ¶106.

[150] Leong, 5:28-32; Weissman, ¶107.

[151] *See* Leong, 5:61-67; Weissman, ¶107.

[152] Weissman, ¶107.

[153] *Id.*

[154] Leong, 5:20-24; Weissman, ¶108.

units inherently disclose the recited agents.[155] A POSITA would have also understood Leong's processing units create the data frames for communicating using the standard Ethernet network.[156]

#### 5. Claim Element 1.4: "wherein the CPU populates the task pool by dividing the requirement into one or more threads and placing the threads in the task pool, each thread comprising one or more tasks, and the matching task being one of the tasks"

Leong discloses this claim element.[157] The '777 Patent describes how "[a] thread 21 is composed of one or more tasks 22."[158] The patent's threads can also describe the order tasks 22 are performed and any conditions that can affect the order.[159] FIG. 2 of the patent depicts the task pool 13 having multiple threads including thread 21A (green) which includes multiple tasks 22A, 22B, 22C, 22D (orange) which has a specific defined order.[160] FIG. 2 of the patent shows how task 22A executes first, and then either task 22B, 22C executes, and finally task 22D if appropriate conditions are met.[161]

---

[155] Weissman, ¶108.

[156] *Id.*

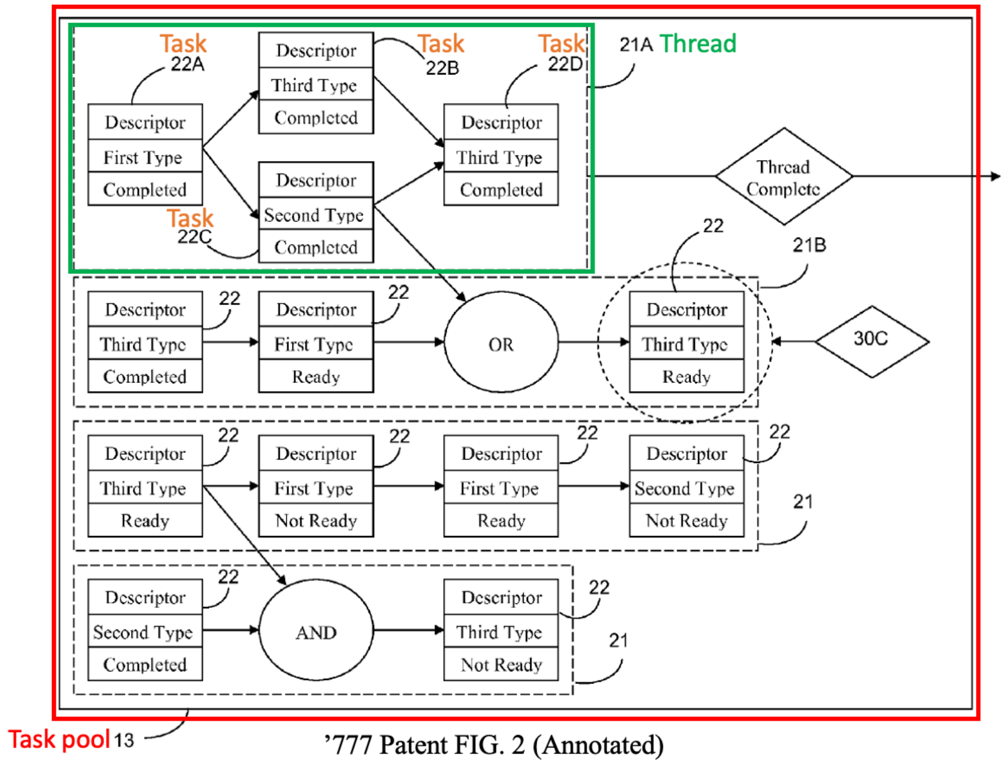
[157] *Id.*, ¶¶110, 118.

[158] '777 Patent, 4:27-28; Weissman, ¶111.

[159] '777 Patent, 4:63-67; Weissman, ¶111.

[160] '777 Patent, 4:67-4; Weissman, ¶111.

[161] Weissman, ¶111.




’777 Patent FIG. 2 (Annotated)

As explained for claim elements 1.2, 1.3, Leong’s processing units, when acting as surveying agents, determine which tasks need to be performed to accomplish the overall project, define the status information for each task, and post the status information to the bulletin board.[162] As discussed for claim element 1.3, after posting the tasks, the processing units read the bulletin board to determine whether any posted task matches their abilities (i.e., are a matching task).[163]

---

[162] *Id.*, ¶112.

[163] *Id.*, ¶112.

Regarding posting tasks to the bulletin board, Leong explains that each task is a "unit of work."[164] The kinds of tasks posted on Leong's bulletin board aren't important.[165] Leong's FIG. 3 depicts a collection of tasks in the bulletin board.[166] Each task includes status information (green) such as the task type, task priority, whether the task is currently being executed, whether the task is complete, and conditions that must be met prior to task execution.[167] Using this status information, Leong's processing units can determine which tasks should be executed, and in what order.[168]

---

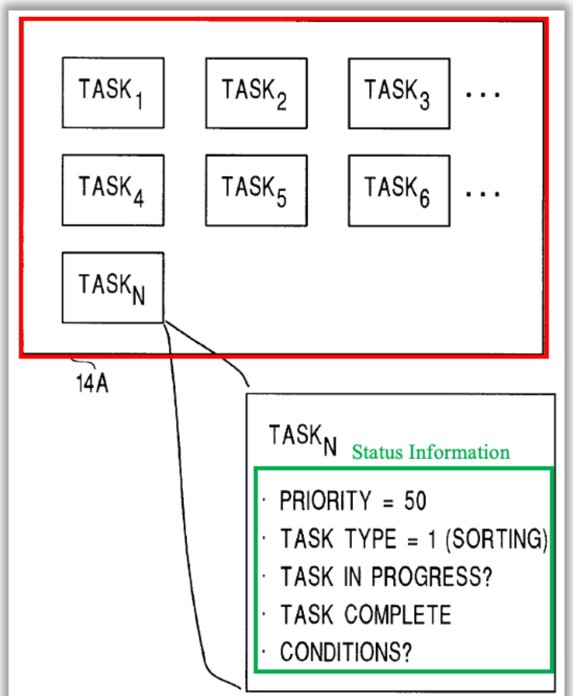
[164] Leong, 2:50-51; Weissman, ¶113.

[165] Leong, 2:54-64; Weissman, ¶113.

[166] Weissman, ¶114.

[167] Leong, 3:19-25; Weissman, ¶114.

[168] Leong, 3:31-40; Weissman, ¶114.




Leong FIG. 3 (Annotated)

Thus, Leong discloses how the task pool is populated by tasks that are partitioned aspects of the overall work and places these tasks in the bulletin board.[169]

Regarding the recited threads, as discussed in Section IV.D.2 above, the usage of "thread" in the '777 Patent isn't consistent with a POSITA's understanding of this term.[170] The claim only requires that "each thread compris[e] one or more

[169] Weissman, ¶115.

[170] *Id.*, ¶116.

tasks."[171] And as discussed in Section IV.D.2, "thread" in the patent means a discrete computational task component of the large computing requirement.[172] This claim element further clarifies how the thread comprises one or more subtasks.[173]

Leong discloses sequences of tasks.[174] Leong explains the status information conditions can define a sequence for execution. For example, Leong describes how Task5 requires Task1 to be completed before execution.[175] A POSITA would have understood Leong's sequences of tasks as disclosures of the claimed threads comprising one or more subtasks because each sequence contains multiple tasks and Leong's tasks are a portion of the overall work.[176]

#### 6. Claim Element 1.5: "wherein each task comprises a descriptor, the descriptor containing at least: a function to be executed; and"

Leong discloses or renders obvious this claim element.[177] Each Leong task has status information, including the task type.[178]

---

[171] *Id.*

[172] *Id.*

[173] *Id.*

[174] *Id.*, ¶117.

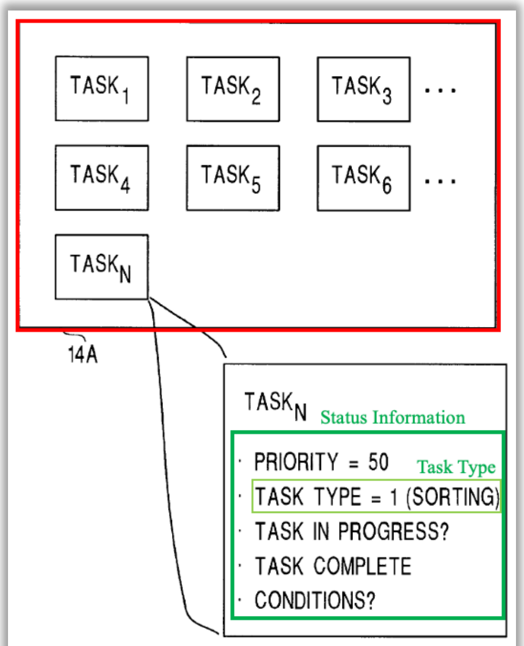
[175] Leong, 3:62-4:3; Weissman, ¶117.

[176] Weissman, ¶117.

[177] *Id.*, ¶¶119, 124, 129.

[178] Leong, 3:19-25, FIG. 3; Weissman, ¶120.

Leong's FIG. 3 highlights the status information (green) with an example task with "sorting" as the type (green).[179]



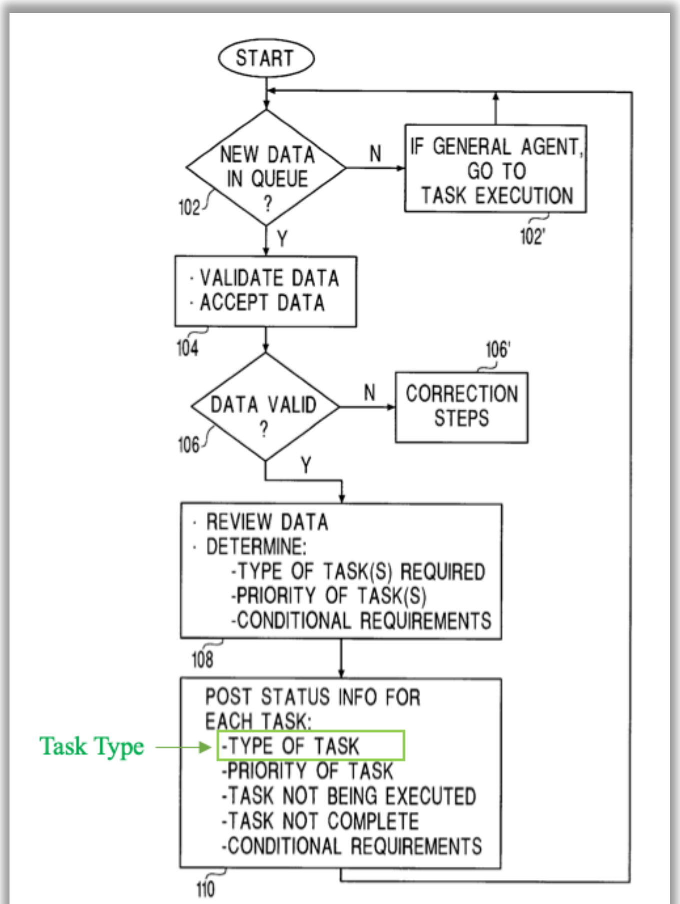

Leong FIG. 3 (Annotated)

A POSITA would have understood Leong's disclosure of a task type stored in the status information as a disclosure of the recited "function to be executed."[180] Leong explains "the application software program running in a particular processing unit 12 may cause the unit to only execute one type of task posted on the

[179] Weissman, ¶121.

[180] *Id.*, ¶122.

bulletin board 14*a*."[181] Leong's task type, therefore, defines what function is executed by the processing units.[182] Leong's FIG. 6 below shows how a surveying agent posts status information for each task, including the task type.[183]



Leong FIG. 6 (Annotated)

[181] Leong, 4:20-23; Weissman, ¶122.

[182] Weissman, ¶122.

[183] *Id.*

Further, Leong's processing units read the bulletin board to find matching tasks "[s]ince an indication of the *type of task to be performed*" (i.e., the function to be executed) is a part of the information posted to the bulletin board.[184]

To the extent it is argued Leong's disclosure of a task type in the status information is insufficient, it would have been obvious to modify Leong's status information to identify a specific function to be executed.[185] A POSITA would have been motivated to modify Leong to provide more granular definitions of the specific tasks.[186] For example, Leong's FIG. 3 discloses the task type as "1 (sorting)." By modifying Leong's status information to include the function to be executed, the resulting Leong system can further define whether the sorting is executed using, for example, a particular algorithm.[187]

A POSITA would have recognized such a modification would have been advantageous because Leong's processing units aren't all the same.[188] Leong explains that certain processing units "may be predisposed to execute a certain type of

---

[184] Leong, 3:26-30; Weissman, ¶123.

[185] Weissman, ¶125.

[186] *Id.*

[187] *Id.*

[188] *Id.*, ¶126.

task."[189] Leong's processing units "need not be of the same hardware or software configuration"[190] A POSITA would have therefore been interested in finding a way to utilize these differences between processing units.[191] And Leong contemplates modifying its status information when explaining that other information may be included in the posted task.[192] A POSITA would have therefore had a reasonable expectation of success when modifying Leong's status information.[193]

With this more specific information, Leong's processing units identify more suitable tasks for execution.[194] If certain Leong processing units perform some tasks more efficiently or are better suited for certain functional implementations, the modified status information would allow these particular Leong processing units to identify and select these tasks.[195] Conversely, if there are certain functions some processing units are unsuitable for executing, the more specific information

[189] Leong, 4:30-35; Weissman, ¶126.

[190] Leong, 6:4-14; Weissman, ¶126.

[191] Weissman, ¶126.

[192] Leong, 3:21 (listing categories of information that *may* be included in the posted task and not excluding others); Weissman, ¶126.

[193] Weissman, ¶126.

[194] *Id.*, ¶127.

[195] *Id.*

would also help these processing units avoid those tasks.[196] If, for example, quick-sort is needed for a particular task, the task's status information, when modified to include a specific function, provides this information prior to execution.[197] The quicksort algorithm requires more memory than other algorithms, so some processing units may be unsuitable because of hardware limitations or because of other applications executing and consuming memory. Because modified status information specifies the function, these Leong processing units avoid selecting this task.[198] And the processing units suited for executing this function could then select this task.[199]

A POSITA would have therefore been motivated to modify Leong's status information so that this more granular information is included and considered when Leong's processing units are identifying tasks.[200]

196 *Id.*

197 *Id.*

198 *Id.*

199 *Id.*

200 *Id.*, ¶128.

#### 7. Claim Element 1.6: "a memory location of data upon which the function is to be executed;"

Leong discloses or renders obvious this claim element.[201] Leong stores the data in memory until a processing unit is ready to manipulate the data.[202] When a surveying agent has processed newly arrived data, the data is made available for manipulation "by posting the status information" onto the bulletin board.[203] As explained for claim element 1.5, Leong's task status information posted to the bulletin board (i.e., task pool) contains several categories of information.[204] Leong's processing units, when acting as survey agents, make the data available for manipulation when the task is posted to the bulletin board.[205] A POSITA would have therefore understood the surveying agents have identified the data's location.[206]

---

[201] *Id.*, ¶¶130, 133.

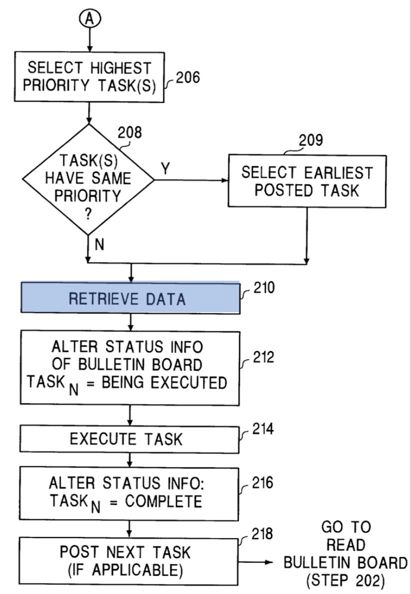
[202] Leong, 4:4-8 ("[T]he electronic data preferably enters the memory 14 which acts as a queue for storing the data until a micro-processing unit 12 can manipulate the data."); Weissman, ¶131.

[203] Leong, 5:51-56; Weissman, ¶131.

[204] Weissman, ¶131.

[205] Leong, 5:53-56 ("Once validated, sequence 100 makes the electronic data available for manipulation by posting the status information for identified tasks onto the bulletin board 14a."); Weissman, ¶131.

[206] Weissman, ¶131.




Leong FIG. 7C (Annotated)

Leong's FIG. 7C shows the processing units retrieving the data necessary to perform the task from the memory after locating a matching task on the bulletin board (step 210, blue).[207] In both instances, Leong's processing units must know the data's location.[208] And although it is possible for a separate location, i.e., not as part of Leong's status information, to store the locations of data yet to be processed, such an implementation would be a centralization of control and contrary to Leong's teachings of there being "no central management and/or control

[207] Leong, 7:40-42 ("At step 210 the electronic data corresponding to the selected task, for example TASKn, is retrieved (if necessary) from the memory.") Weissman, ¶132.

[208] Weissman, ¶132.

unit."[209] A POSITA would have therefore understood Leong's tasks to inherently disclose the processing unit receiving the data's memory location.[210]

Thus, Leong discloses that each task descriptor also contains a memory location of data upon which the function is to be executed.[211]

To the extent it is argued Leong's disclosure is insufficient, it would have been obvious to a POSITA to include the memory location of data upon which the function is to be executed in Leong's status information.[212] In Leong's system, the status information provides the information needed for the processing agents to identify appropriate tasks and to execute those tasks.[213] To execute the tasks, the processing units must know the data's location.[214] And including this information in Leong's status information would have been obvious so that upon retrieval of

---

[209] Leong, 2:47-48; Weissman, ¶132.

[210] Weissman, ¶132.

[211] *Id.*, ¶133.

[212] *Id.*, ¶134.

[213] *Id.*

[214] *Id.*

the task's status information, Leong's processing units would have this key information available.[215] And this would be consistent with Leong's teachings against central management or control.[216]

#### 8. Claim Element 1.7: "wherein the first agent is a data frame comprising: a source address, a destination address and a payload;"

Leong discloses this claim element.[217] As explained for claim elements 1.1, 1.2, Leong discloses generalized and specialized processing units connected using Ethernet.[218] For the different aspects of Leong's system to communicate using Ethernet, a POSITA would have found it inherent that Leong's agents utilize data frames which are essential and necessary parts of standard Ethernet networks.[219] The '777 Patent also states the agents "may be considered a data frame in the networking sense."[220]

It would not be possible for a system to use standard Ethernet networks without utilizing Ethernet data frames.[221] A data frame is part of layer 2 in the

---

[215] *Id.*

[216] *See* Leong, 2:47-48; Weissman, ¶134.

[217] Weissman, ¶¶135, 139.

[218] *See*, Leong, 5:20-24, 6:10-14; Weissman, ¶136.

[219] Weissman, ¶136.

[220] '777 Patent, 5:28-32; Weissman, ¶136.

[221] Ethernet, 12, 14, 30-32; Weissman, ¶137.

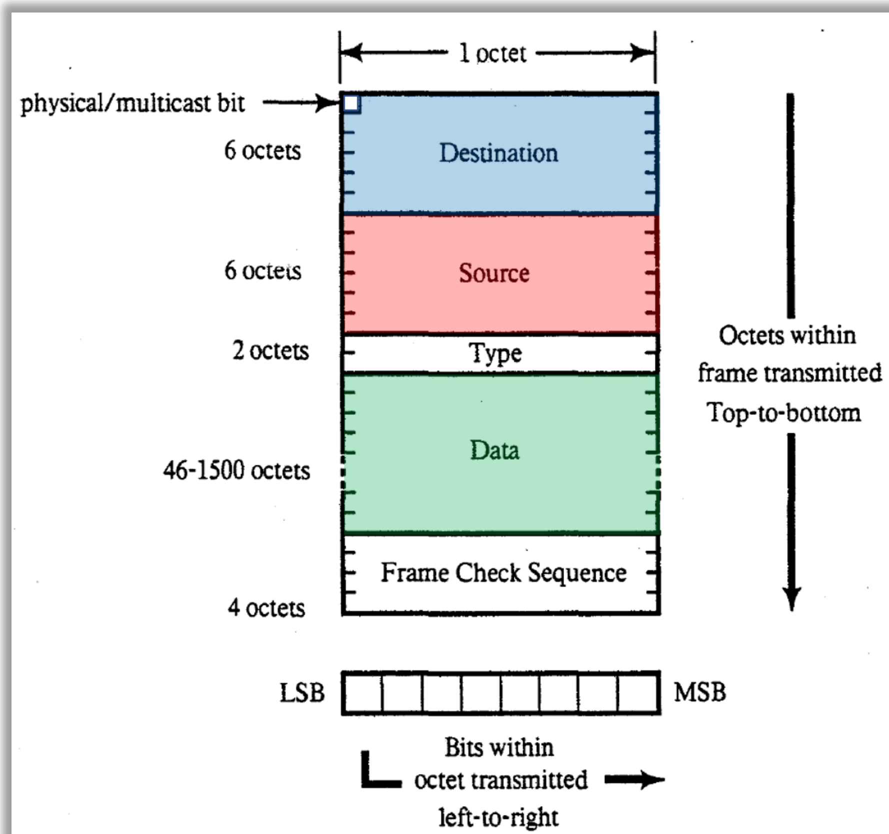
Open Systems Interconnection ("OSI") model and is fundamental to the operation of standard networks including Ethernet networks.[222] A POSITA would have been well aware that Ethernet data frames include a source address, a destination address, and a payload, and would have also been aware of their usage.[223] In the Ethernet standard's first version, published in 1980, the Ethernet frame format was defined.[224] The Ethernet standard depicts the five fields in Figure 6-1, reproduced below, including source address (pink), destination address (light blue), and data (light green).[225]

[222] Weissman, ¶137.

[223] Ethernet, 30-32; Weissman, ¶137.

[224] Weissman, ¶137.

[225] Ethernet, 19-21; *see also* Kurose, 471-473, Figure 5.22; Weissman, ¶137.




"Ethernet" Figure 6-1 (Annotated)

Thus, a POSITA would have understood Leong's processing units, connected through an Ethernet network, to inherently use the necessary underlying mechanisms of a standard Ethernet network, including an Ethernet data frame.[226] And because the standard Ethernet data frame includes source address, destination address, and data fields with specifically defined purposes, a POSITA would have understood Leong discloses this claim element.[227]

#### 9. Claim Element 1.8: "wherein the first agent retrieves the matching task by: being dispatched by the first solidarity

[226] Weissman, ¶138.

[227] Ethernet, 30-32; Weissman, ¶139.

**cell to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the first solidarity cell is configured to perform;"**

Leong discloses or alone renders obvious this claim element.[228] As discussed for claim element 1.3, Leong discloses processing units (i.e., solidarity cells) that read the bulletin board (i.e., task pool) to find matching tasks to process.[229] And as explained for claim element 1.7, since Leong's processing units are coupled over an Ethernet network, a POSITA would have understood that Leong inherently discloses the use data frame fields, including the source address, destination address and payload, in a conventional manner.[230] This is necessary for Leong's processing units to participate in an Ethernet network.[231]

Because Leong's processing unit is the source and the bulletin board is the destination, and because Leong's units are coupled over an Ethernet network, a POSITA would have understood Leong also inherently discloses using the data frame in a manner where the source address is set to the processing unit's address, and the destination address is set to the bulletin board's address.[232] This standard

---

[228] Weissman, ¶¶140, 144.

[229] *Id.*, ¶140.

[230] *Id.*

[231] Leong, 5:20-24; Weissman, ¶140.

[232] Weissman, ¶141.

usage of both the source and destination address fields is necessary so that Leong's processing units can participate in a standard Ethernet network.[233]

Further, because Leong describes how the processing units can identify appropriate tasks, a POSITA would also understand the data frame payload would include functions that can be performed by the processing units.[234] As discussed for claim element 1.3, Leong's processing units identify appropriate tasks for execution using the status information of each task.[235] To implement this aspect described by Leong, a POSITA would have avoided using a central listing that would specify the tasks a particular Leong processing unit could perform because such a central listing would be contrary to Leong's teachings of decentralization.[236] In this instance, matching tasks in Leong's bulletin board are sought, so this information should be included with the data frame.[237] Doing so would allow the destination to immediately execute appropriate tasks.[238] Storing this information in the data frame

[233] *Id.*

[234] *Id.*, ¶142.

[235] *See* Leong, 3:26-4:3 (describing status information for each task), 4:18-35 (describing specialized and general agents), 6:47-61 (describing FIG. 7A); Weissman, ¶142.

[236] *See* Leong, 2:47-48; Weissman, ¶142.

[237] Weissman, ¶142.

[238] *Id.*

payload would be consistent with the standard usage of this field of the Ethernet data frame and necessary for participation in a standard Ethernet network.[239]

To the extent it is argued this disclosure is insufficient, it would have been obvious to modify Leong so the Ethernet data frame payload includes the information needed to identify the tasks so that the destination can execute appropriate tasks.[240] In this instance, matching tasks in Leong's bulletin board with the processing units' capabilities is desired so this information should be included with the data frame.[241] Utilizing the Ethernet data frame payload would be consistent with a POSITA's understanding of the data frame fields.[242] And including this information in the data frame would improve the efficiency of task matching while being consistent with Leong's teachings.[243] This is because the information needed to find the appropriate task would be included with the very data frame that is triggering the search for an appropriate task and there wouldn't be any central management or control.[244]

---

[239] *Id.*

[240] *Id.*, ¶143.

[241] *Id.*

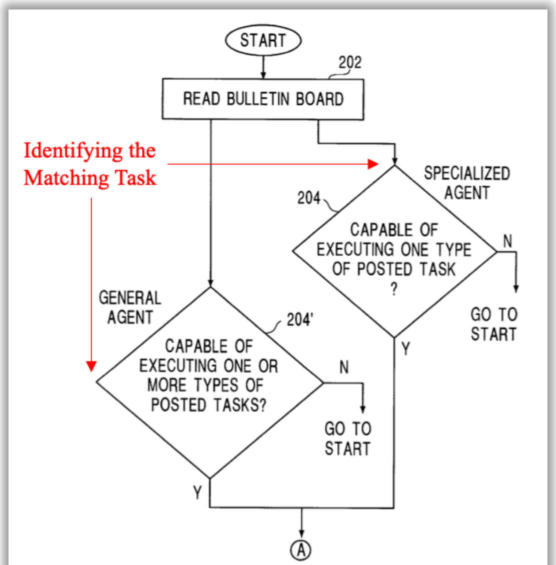
[242] *Id.*

[243] *Id.*

[244] *See* Leong, 2:47-48; Weissman, ¶143.

### 10. Claim Element 1.9: "searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform; and"

Leong discloses or renders obvious this claim element.[245] As discussed for claim element 1.3, Leong discloses processing units (i.e., solidarity cells) that read the bulletin board (i.e., task pool) to find matching tasks to process.[246] FIG. 7A shows decisions by both a specialized agent (step 204) or general agent (step 204') in choosing a task to perform.[247]




Leong FIG. 7A (Annotated)

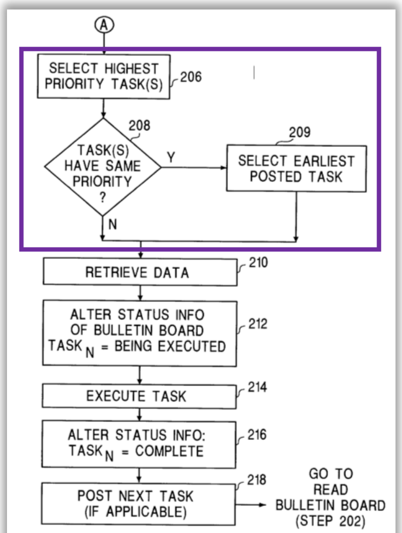
---

[245] Weissman, ¶¶145, 150.

[246] *Id.*, ¶146.

[247] *Id.*

In addition, Leong's processing units determine if the matching task is ready to be processed by reviewing the task priority (steps encircled in purple in Fig. 7C below).[248]




Leong FIG. 7C (Annotated)

Further, Leong's processing units examine whether the status information indicates the task is being processed.[249] And Leong's processing units also examine whether the status information indicates any conditions must be met before the task is executed.[250] Once these checks are complete, Leong's processing units will consider the task ready for processing.[251]

[248] Leong, 3:32-41, FIG. 7C (elements 206, 208, 209); Weissman, ¶147.

[249] Leong, 3:45-61; Weissman, ¶148.

[250] Leong, 3:62-67; Weissman, ¶148.

[251] Weissman, ¶148.

And as discussed with claim element 1.5, Leong discloses or renders obvious how the descriptor contains at least a function to be executed.[252] When matching tasks, Leong's processing units would therefore consider whether they included the appropriate function.[253]

#### 11. Claim Element 1.10: "returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address, and the payload comprises the descriptor of the matching task."

Leong discloses or alone renders obvious this claim element.[254] As explained for claim element 1.8, since Leong's processing units are coupled over an Ethernet network, a POSITA would have understood that Leong inherently discloses the necessary use of the standard's data frame fields, including source address, destination address and payload, in a typical manner.[255]

Because Leong's bulletin board is the source and the processing unit is the destination, a POSITA would have understood Leong inherently discloses using the data frame in a manner where the source address is set to the bulletin board's

[252] *Id.*, ¶149.

[253] *Id.*

[254] *Id.*, ¶¶151, 154.

[255] *Id.*, ¶151.

address, and the destination address is set to the processing unit's address.[256] The data frame payload would include the necessary information for the destination to execute appropriate tasks.[257] In this instance, the matching tasks from Leong's bulletin board are returned to the solidarity cell, so this information is included with the data frame.[258] And as discussed for claim element 1.8, a POSITA would have recognized the usage of the Ethernet data frame in this standard manner is necessary for participation in a standard Ethernet network.[259]

To the extent Leong's disclosure is insufficient, it would have been obvious to include the descriptor of the matching task with the data frame payload so the solidarity cell can begin execution of the matching task upon receipt.[260] Including this information in the data frame would be efficient because Leong's processing unit could then access the data frame payload and begin the process of task execution, without having to consult a centralized resource.[261] The information needed to begin task execution would be included with the data frame from Leong's bulletin

[256] *Id.*, ¶152.

[257] *Id.*, ¶152.

[258] *Id.*

[259] *Id.*

[260] *Id.*, ¶153.

[261] *Id.*

board, and no central management or control is necessary.[262] Such a modification would have a reasonable expectation of success because it would utilize the Ethernet data frame in a standard manner.[263]

For all the reasons explained above, Leong renders obvious claim 1.[264]

### C. Dependent Claim 2: "The apparatus of claim 1 wherein the task pool notifies the CPU when the tasks of a thread are completed."

Leong alone renders obvious the additional element recited in this claim.[265]

As discussed above for claim 1's preamble and claim elements 1.1, 1.2, Leong discloses processing units associated with a bulletin board, and how these units can operate as a "surveying" agent that reviews the data queue for new data, determines the tasks necessary for the overall project, sets the status information for those tasks, and posts them to the associated bulletin board (i.e., task pool).[266] And as discussed for claim element 1.5, Leong discloses status information for tasks.[267]

Leong's status information indicates when the task is executing and when

---

[262] *See* Leong, 2:47-48; Weissman, ¶153.

[263] Ethernet, 30-32; Weissman, ¶153.

[264] Weissman, ¶155.

[265] *Id.*, ¶¶156, 165.

[266] *Id.*, ¶157.

[267] *Id.*

the task is complete.[268] After performing a task, Leong's processing units change the task status information to "complete" (step 216, light green) to prevent task duplication by other processing units.[269] Leong also explains the status information can impose a sequencing condition that Task1 completes before Task5 executes.[270] In other configurations, Leong's processing units post a new task (step 218, light blue) upon completion of a task.[271] Leong also explains its system is failure proof and all tasks are eventually executed, even if some of the processing units fail or become disconnected.[272]

---

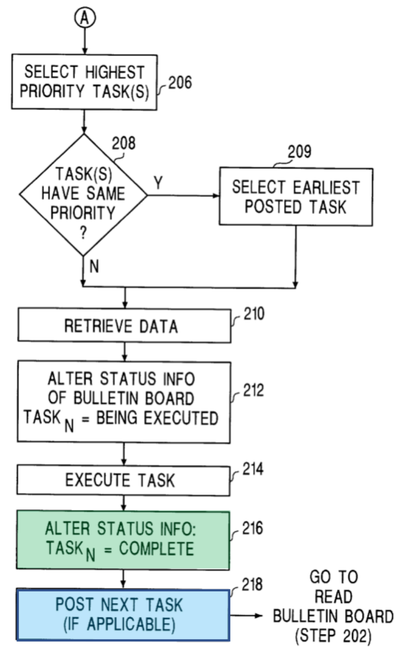
[268] Leong, 3:62-67, 6:42-46; Weissman, ¶158.

[269] Leong, 3:55-61, FIG. 7C (step 216); Weissman, ¶158.

[270] Leong, 3:64-67; Weissman, ¶158.

[271] Leong, 7:46-52; Weissman, ¶158.

[272] Leong, 5:26-35; Weissman, ¶158.




Leong FIG. 7C (Annotated)

A POSITA would have understood that Leong's processing units' change in status information to "complete" on the bulletin board (i.e., task pool) notifies processing units, including surveying agents (i.e., CPUs).[273] Leong's processing units review the status information of tasks in the bulletin board, including the status information for completed tasks.[274] Leong's bulletin board notifies the processing units of task completion because the status information of tasks is updated to reflect task completion, and the bulletin board holds the status information, and the

---

[273] Weissman, ¶159.

[274] Leong, 3:52-55 ("Accordingly, when the micro-processing units 12 read the bulletin board 14a, they will not execute that task because it is already being executed by another agent"), 3:64-67, 7:46-52; Weissman, ¶159.

processing units continually read the status information.[275] And with task completion information, Leong's system executes sequences where subsequent tasks or new tasks are executed upon a prior tasks' completion.[276] Leong's system can also ensure all electronic data is processed by monitoring task completion information.[277]

To the extent it's argued Leong's disclosures are insufficient, it would have been obvious to a POSITA to modify Leong to implement task completion notifications where the task pool notifies the CPU of completion.[278] A POSITA would have been motivated to modify Leong and have a reasonable expectation of success because Leong expressly discloses task sequencing, which requires some form of notification of task completion.[279] Leong contemplates executing another task after a first task completes, and creating a new task after another task completes.[280] Both are events occurring subsequent to task completion.[281] Task completion notifications would also provide information for Leong's monitor 32, which provides

[275] Weissman, ¶159.

[276] *Id.*

[277] *Id.*

[278] *Id.*, ¶160.

[279] *Id.*

[280] Leong, 3:64-67, 7:46-52; Weissman, ¶160.

[281] Leong, 3:64-67, 7:46-52; Weissman, ¶160.

"comprehensive system information including agent status, batch status, error status, event status, and/or process bottlenecks."[282] At a minimum, a POSITA would have found the task sequencing Leong contemplates would require some form of task completion notification.[283] In addition to Leong's changing of the status information to "complete" discussed above, a POSITA would have found it obvious to further modify Leong to implement task notifications where Leong's bulletin board provides a notification to processing units, including when a processing unit operates a surveying agent.[284]

Another reason a POSITA would have been motivated to implement a task completion notification is because of Leong's failure proof aspirations.[285] If one or more processing units "were to become disconnected from the network or become incapable of performing tasks," Leong's system would nevertheless process the data.[286] Leong explains other processing units "would simply execute the posted tasks as designed."[287] Leong doesn't elaborate on how the system would know the

---

[282] Leong, 10:46-51; Weissman, ¶160.

[283] Weissman, ¶160.

[284] *Id.*

[285] Leong, 5:29-35; Weissman, ¶161.

[286] Leong, 5:26-35; Weissman, ¶161.

[287] Leong, 5:26-35; Weissman, ¶161.

tasks did not or would not complete. But one obvious way a POSITA would have ensured task execution is to implement a task timeout.[288] If a task completion notification wasn't received within a certain period of time after task execution begins, the system would presume the execution of the task had failed.[289] In addition to ensuring every task is executed, a timeout for each task ensures a few failed tasks won't prevent the larger task from completing within a certain timeframe or prevent the larger task from executing at a certain throughput. For example, Leong's example system processes check images.[290] A POSITA would have understood specifying a timeout would help ensure the system processes checks at a certain throughput.[291] This technique would have been well within the skill of a POSITA to implement in Leong especially because task status information already includes a field indicating a task is executing.[292]

When a processing unit claims a task, Leong's status information is updated to indicate task execution.[293] If the task completes normally, the processing unit

[288] Weissman, ¶161.

[289] *Id.*

[290] *See* Leong, 8:8-10:51, FIG. 8; Weissman, ¶161.

[291] Weissman, ¶161.

[292] *See* Leong, FIG. 3, 6:36-46; Weissman, ¶161.

[293] Leong, 6:42-46; Weissman, ¶162.

modifies the status information to indicate completion.[294] In the modified Leong system, if a period of time elapses where the task's status information isn't modified to indicate completion, the system presumes the task has failed and needs reassignment.[295] Processing units including those acting as surveying agents could, as they read through the tasks in the bulletin board, respond to the timeout by modifying the task status information so task execution is no longer indicated and available processing units will execute the task.[296] Thus, Leong's failure proof intentions would have suggested to a POSITA that a task completion notification scheme should be implemented.[297]

And finally, a POSITA would have recognized task completion notifications would help manage finite memory resources.[298] Leong's bulletin board resides in memory, and the adding of items to the bulletin board is described.[299] But Leong doesn't describe what happens to tasks in the bulletin board after the tasks have completed execution. [300] A POSITA would have recognized implementing task

---

[294] Leong, 6:42-46; Weissman, ¶162.

[295] Weissman, ¶162.

[296] *Id.*

[297] *Id.*

[298] *Id.*, ¶163.

[299] Leong, 6:18; Weissman, ¶163.

[300] Weissman, ¶163.

completion notifications in Leong would facilitate periodic removal of old status information for completed tasks and would have recognized the memory as a finite resource and one of the centralizations in the system.[301] Otherwise, memory usage would continually grow due to unremoved status information.[302] A POSITA would have recognized this issue when implementing Leong's teachings and taken appropriate steps.[303] Implementing a task completion notification allows any memory management technique to know which items need not remain in memory.[304] In the modified system, Leong's surveying agents or other processing units perform this removal for a certain period after receiving a task completion notification.[305] Leong's surveying agents already process the items in memory and could execute this removal as part of other housekeeping tasks.[306] Other Leong processing units also review the status information in the bulletin board and execute this removal while reading through the different items in the bulletin board.[307] Another possible implementation is dedicating certain processing units, such as surveying agents, to

---

[301] *Id*

[302] *Id*.

[303] *Id*.

[304] *Id*.

[305] *Id*.

[306] Leong, 4:9-16; Weissman, ¶163.

[307] Leong, 3:52-55, 3:64-67, 7:46-52; Weissman, ¶163.

remove status information.[308] Doing so would make surveying agents the only units to add and remove status information and would be an obvious implementation choice.[309] And modifying Leong's processing units to remove unneeded task information based on task completion notifications would have been desirable so that long completed tasks don't needlessly consume finite memory resources.[310]

For at least these reasons, it would have been obvious to a POSITA to modify Leong to implement task completion notifications.[311] Leong's disclosure of events occurring after task completion would necessitate some sort of task completion notification.[312] Leong's emphasis on a failure proof system similarly requires monitoring task status to complete all tasks in a timely fashion.[313] And a POSITA would recognize the need to manage finite memory resources and use task completion notifications to implement a form of garbage collection to reclaim memory from old tasks.[314]

### D. Dependent Claim 3: "The apparatus of claim 1 wherein the tasks each comprise a task type selected from a set of task types, and

[308] Weissman, ¶163.

[309] *Id.*

[310] *Id.*

[311] Weissman, ¶164.

[312] *Id.*

[313] *Id.*

[314] *Id.*

**wherein the first solidarity cell is configured to perform tasks of one or more of the task types."**

Leong discloses the additional elements recited in this claim.[315] As discussed above for claim element 1.4, Leong explains that the overall work is divided into smaller tasks and that a POSITA would have understood how to do so.[316] Leong further teaches specific task types included as part of the overall project in the context of its disclosed embodiment.[317] And these tasks can be partitioned into "accepting tasks," "sorting tasks," and "distributing tasks."[318]

Leong also teaches the processing units are specialized agents (able to perform only one task type) or general agents (configured to perform one or more task types).[319]

---

[315] Weissman, ¶¶166, 168.

[316] *Id.*, ¶166.

[317] Leong, 2:57-62, 3:6-11 (identifying task types posted on the electronic bulletin board). *See also*, *id.*, 8:27-32 (naming the six basic agents that process the tasks required in the disclosed embodiment's overall project), 9:55-10:16 (discussing the task types each of the agents perform in the disclosed embodiment); Weissman, ¶166.

[318] Leong, 2:57-62; Weissman, ¶166.

[319] Leong, 4:17-29, FIGS. 7A; Weissman, ¶167.

### E. Dependent Claim 4: "The apparatus of claim 3 wherein the matching task is a task that is ready to be processed and has a task type that the first solidarity cell can perform."

Leong discloses the additional elements recited in this claim.[320] As discussed for claim element 1.3, Leong discloses processing units (i.e., solidarity cells) reading the bulletin board (i.e., task pool) to find matching tasks to process.[321] And as discussed for claim element 1.9, Leong teaches that the processing units determine if the matching task is ready for processing.[322]

### F. Dependent Claim 5: "The apparatus of claim 4 wherein the first agent retrieves the matching task by: searching the task pool for a task that is ready to be processed and has a task type that the first solidarity cell can perform; and identifying the matching task."

Leong discloses the additional elements recited in this claim.[323] As discussed for claim elements 1.3, 1.9, and claim 4, Leong discloses processing units (i.e., solidarity cells) reading the bulletin board (i.e., task pool) to find matching tasks that are ready to process.[324]

---

[320] Weissman, ¶¶169, 171.

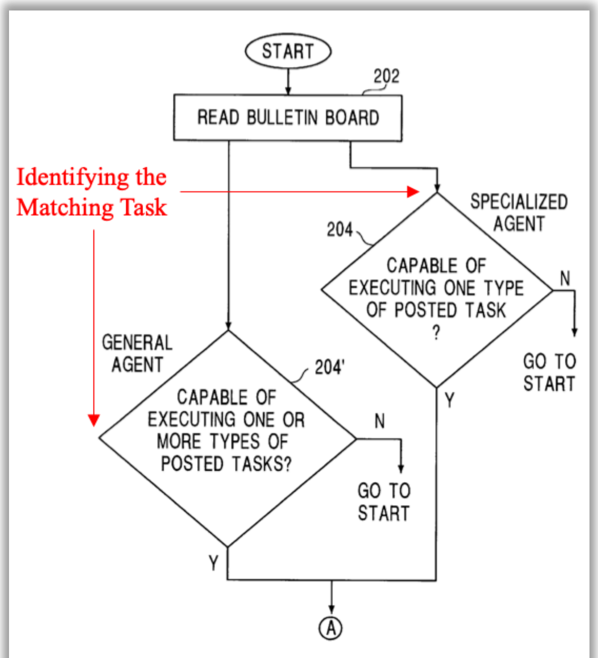
[321] *Id.*, ¶170.

[322] *Id.*, ¶170.

[323] *Id.*, ¶¶172, 174.

[324] *Id.*

When Leong's processing units read the bulletin board, they identify whether any of the posted tasks match their abilities.[325]



Leong FIG. 7A (Annotated)

### G. Dependent Claim 6: "The apparatus of claim 1 wherein the matching task is a task that is ready to be processed and the function of the matching task can be performed by the first solidarity cell."

Leong discloses the additional elements recited in this claim.[326] As discussed for claim elements 1.3, 1.9 and dependent claim 4, Leong discloses processing

[325] Leong, 6:52-61, FIG. 7A; Weissman, ¶173.

[326] Weissman, ¶¶175, 177.

units (i.e., solidarity cells) reading the bulletin board (i.e., task pool) to find matching tasks (i.e., the function can be performed by the solidarity cell) that are ready to process.[327]

### H. Dependent Claim 7: "The apparatus of claim 6 wherein the first agent retrieves the matching task by searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform, and identifying the matching task."

Leong discloses the additional elements recited in this claim.[328] As discussed for claim elements 1.3, 1.9 and dependent claim 5, Leong discloses a processing unit (i.e., agent) searching the bulletin board (i.e., task pool) for a task that is ready to be processed, with a task type that it can perform, and identifying the matching task.[329]

### I. Dependent Claim 8: "The apparatus of claim 1 wherein the descriptor further contains a memory location where processed data is to be stored."

Leong discloses or renders obvious the additional elements recited in this claim.[330]

---

[327] *Id.*, ¶176.

[328] *Id.*, ¶¶178, 180.

[329] *Id.*, ¶179.

[330] *Id.*, ¶¶181, 186.

Leong discloses its system includes a device for making the manipulated data available to the client.[331] When describing the image distribution and support system, Leong explains that CD scheduling and microfilm scheduling agents "monitor the bulletin board 14a" waiting for specific volume thresholds before scheduling the task.[332]

Although Leong doesn't expressly describe how the manipulated data reaches the output device, the CD scheduling, or the microfilm scheduling agents, a POSITA would have understood Leong's processing units would need to deliver the manipulated electronic data.[333] Because Leong describes how the output device receives the manipulated electronic data, a POSITA, when considering Leong's disclosure, would have found it inherent that a memory location for storage of the manipulated data would be needed, and that such a memory location would be defined with the task as part of Leong's status information.[334] A POSITA would recognize this would be necessary and critical when jobs involve tasks manipulating

[331] Leong, 5:38-45; Weissman, ¶182.

[332] Leong, 10:5-16; Weissman, ¶183.

[333] Weissman, ¶184.

[334] *Id.*

data larger than available memory in Leong's processing units, requiring storage for intermediate results as the processing unit executes the task.[335]

To the extent this isn't found to be inherent, storing a memory location in Leong's status information would have been obvious because by doing so, each of Leong's processing units wouldn't need to contact another aspect of the system to determine where the processed data is to be stored.[336] Instead, each of Leong's processing units would have this information available through the already acquired status information.[337] In situations where a sequence of tasks are executed, Leong's processing units would consult the descriptor instead of contacting another aspect of the system to obtain the processed data's location.[338] Even when a single task is executed, including this information allows a function to itself store the output at the specific location as execution continues.[339] This would be particularly desirable when the output is larger than the processing unit's storage capacity.[340] And such a design would also be consistent with Leong's decentralization teachings.[341]

---

[335] *Id.*

[336] *Id.*, ¶185.

[337] *Id.*

[338] *Id.*

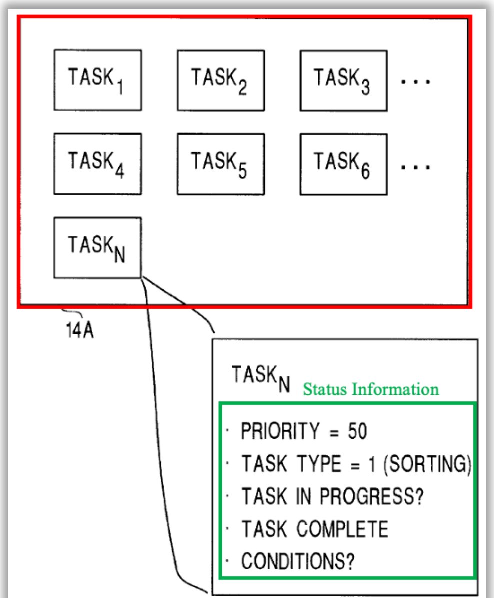
[339] *Id.*

[340] *Id.*

[341] *See* Leong, 2:47-48; Weissman, ¶185.

### J. Dependent Claim 9: "The apparatus of claim 1 wherein the descriptor is a data structure and the task contains a reference to the memory location of the descriptor."

Leong discloses or renders obvious the additional elements recited in this claim.[342] As explained for claim element 1.5, each Leong task has status information.[343] As depicted by Leong's FIG. 3, reproduced below, the status information for each task includes priority, task type, whether the task is in progress, whether the task is complete, and if there are any execution conditions.[344]




Leong FIG. 3 (Annotated)

A POSITA would have understood Leong's status information (which as discussed for claim element 1.5 is a disclosure of the recited descriptor) is a data

[342] Weissman, ¶¶187, 191.

[343] *Id.*, ¶188.

[344] Leong, 3:19-25; Weissman, ¶188.

structure because it is a collection of typed data values that all have a relationship to a certain task.[345] To associate the task with this data structure, a POSITA would have understood the task to include a reference to that data structure, avoiding duplication of data in memory.[346] Tasks using the status information—and other portions of the system—would access the same data structure in memory.[347]

To the extent these additional features aren't found to be inherent, it would have been obvious to a POSITA to implement Leong's status information using a data structure that's referenced by the tasks because it applies well-known and widely used programming techniques in a typical manner.[348] As discussed, arranging Leong's status information in a data structure would have been obvious to a POSITA since each value relates to the associated task, and associating such related information with each other in a particular data structure would facilitate access of the information.[349] Associating the Leong task to the data structure of status

[345] Weissman, ¶189.

[346] *Id.*

[347] *Id.*

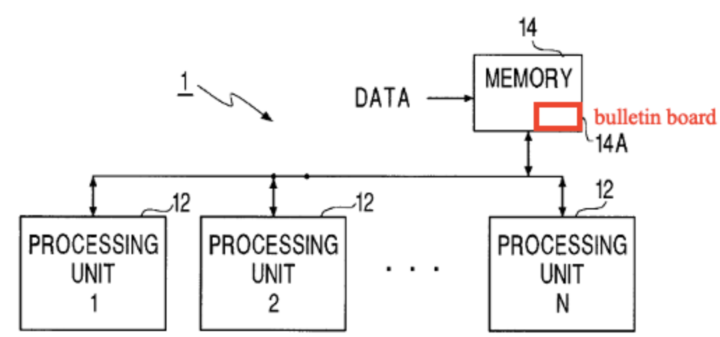
[348] *Id.*, ¶190.

[349] *Id.*

information by reference is an efficient manner of tying Leong's task to the data structure without requiring the duplication of the data structure's values.[350]

### K. Dependent Claim 10: "The apparatus of claim 1 wherein the task pool occupies a region of physical memory."

Leong discloses the additional element recited in this claim.[351] Leong explains the bulletin board (i.e., task pool) resides in memory 14.[352] And Leong's FIG. 2 depicts the bulletin board as a part of the memory.[353]




Leong, FIG. 2 (Annotated)

[350] *Id.*

[351] *Id.*, ¶¶192, 194.

[352] Leong, FIG. 2, 3:9-11, FIG 4 (at element 14A); Weissman, ¶193.

[353] Weissman, ¶193.

### L. Dependent Claim 11: "The apparatus of claim 10 wherein the task pool is disposed in a hardware block dedicated to the task pool."

Leong discloses or renders obvious the additional element recited in this claim.[354]

As explained above for dependent claim 10, Leong's bulletin board (i.e., task pool) resides in memory.[355] Leong depicts the bulletin board as part of the memory, which is a hardware block.[356]

To the extent it is argued Leong doesn't indicate the bulletin board's portion of the memory is dedicated, it would have been obvious to ensure the bulletin board occupies a contiguous portion of memory to facilitate access.[357] Dedicating a portion of Leong's memory would have been well within the skill of a POSITA at the time.[358] A POSITA would have understood how to identify certain ranges of the memory and use those specific ranges for a particular purpose, such as Leong's bulletin board.[359]

---

[354] *Id.*, ¶¶195, 198.

[355] *Id.*, ¶196.

[356] Leong, FIG. 2 (element 14a); Weissman, ¶196.

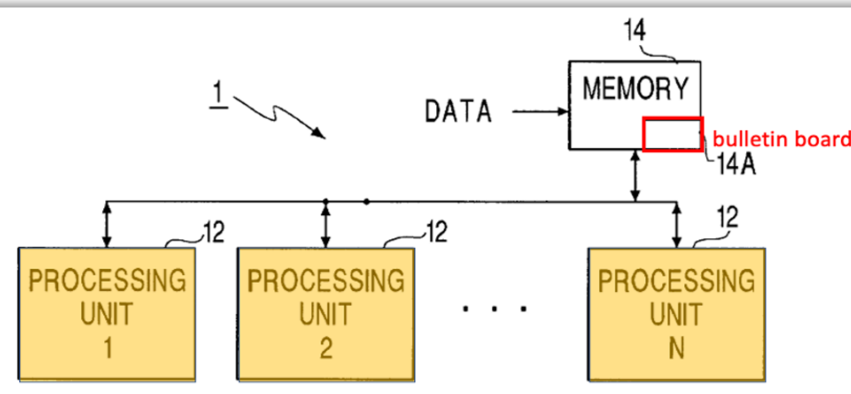
[357] Weissman, ¶197.

[358] *Id.*

[359] *Id.*

### M. Dependent Claim 12: "The apparatus of claim 1 further comprising a second solidarity cell comprising a second agent that proactively retrieves matching tasks from the task pool for the second solidarity cell to process, wherein the matching task for each solidarity cell is a task in the task pool that is ready to be processed and can be performed by the solidarity cell."

Leong discloses the additional elements recited in this claim.[360] As discussed above for claim 1's preamble, claim element 1.3, and dependent claim 4, Leong discloses a system with multiple processing units, each configured to proactively read the bulletin board using agents (i.e., data frames) to locate tasks it can perform.[361] Leong's FIG. 2 depicts multiple processing units and the bulletin board.[362]




Leong FIG. 2 (Annotated)

And as discussed for claim element 1.9, Leong teaches that each processing unit determines if the matching task is ready to be processed. [363]

---

[360] *Id.*, ¶¶199, 202.

[361] *Id.*, ¶200.

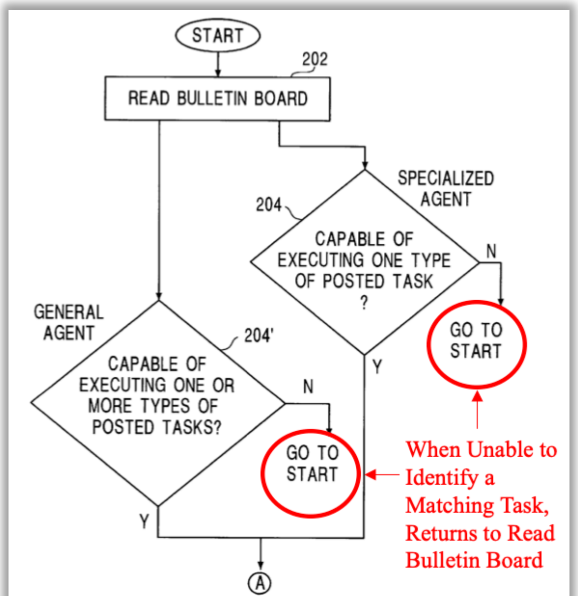
[362] *Id.*

[363] *Id.*, ¶201.

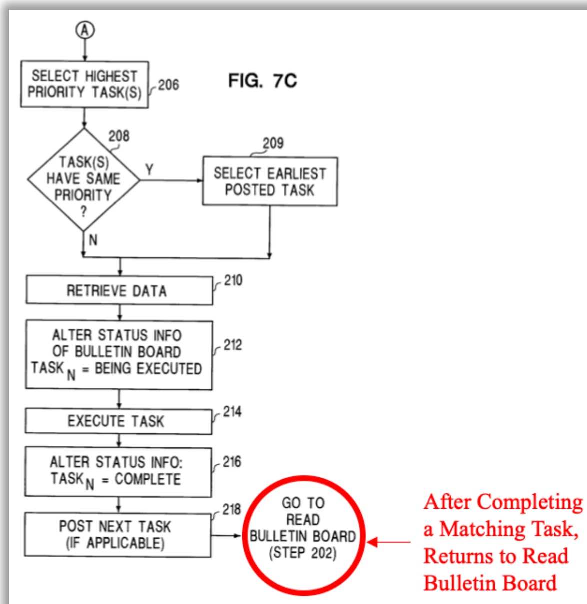
### N. Dependent Claim 13: "The apparatus of claim 12 wherein each solidarity cell sends its agent to the task pool when the solidarity cell does not have a matching task to process."

Leong discloses the additional elements recited in this claim.[364] As discussed for claim element 1.3, Leong teaches that its processing units are proactive and read the bulletin board to search for tasks that match their abilities.[365]

As Leong's FIGS. 7A, 7C show that when a processing unit cannot locate a matching task or has completed a matching task, it proactively returns to read the bulletin board to locate a matching task.[366]




Leong FIG. 7A (Annotated)




Leong FIG. 7C (Annotated)

[364] *Id.*, ¶¶203, 206.

[365] *Id.*, ¶204.

[366] Leong, FIGS. 7A, 7C, 6:60-61, 7:51-52; Weissman, ¶205.

### O. Independent Claim 14

Independent claim 14 is rendered obvious by Leong.[367]

#### 1. Claim 14's Preamble: "An apparatus for parallel processing of a large computing requirement, the apparatus comprising:"

Leong discloses all elements of this preamble to the extent this preamble limits the scope of claim 14.[368] Claim 14's preamble is identical to claim 1's preamble and that discussion is incorporated-by-reference here.

#### 2. Claim Element 14.1: "a central processing unit ("CPU");"

Leong discloses this claim element.[369] Claim element 14.1 is identical to claim element 1.1 and that discussion is incorporated-by-reference here.

#### 3. Claim Element 14.2: "a task pool in electronic communication with the CPU;"

Leong discloses this claim element.[370] Claim element 14.2 is the same as claim element 1.2 and that discussion is incorporated-by-reference here.

#### 4. Claim Element 14.3: "first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising a first agent configured to proactively retrieve, from

---

[367] Weissman, ¶207.

[368] *Id.*, ¶208.

[369] *Id.*, ¶209.

[370] *Id.*, ¶210.

**the task pool, a matching task for the solidarity cell to process; and"**

Leong discloses this claim element.[371] This element is substantially similar to claim element 1.3 and that discussion is incorporated-by-reference here.

### 5. Claim Element 14.4: "a second solidarity cell comprising a second agent that proactively retrieves matching tasks from the task pool for the second solidarity cell to process, wherein the matching task for each solidarity cell is a task in the task pool that is ready to be processed and can be performed by the solidarity cell;"

Leong discloses this claim element.[372] This element is the same as dependent claim 12. The discussions of claim 1's preamble, claim element 1.9, and dependent claims 4 and 12 are incorporated-by-reference here.

### 6. Claim Element 14.5: "wherein each solidarity cell sends its agent to the task pool when the solidarity cell does not have a matching task to process; and"

Leong discloses this claim element.[373] This element is the same as dependent claim 13 and that discussion is incorporated-by-reference here.

---

[371] *Id.*, ¶211.

[372] *Id.*, ¶212.

[373] *Id.*, ¶213.

#### 7. Claim Element 14.6: "wherein each agent comprises a source address, a destination address, and a payload, and wherein"

Leong discloses this claim element.[374] This element is substantially similar to claim element 1.7 and that discussion is incorporated-by-reference here.

#### 8. Claim Element 14.7: "each agent retrieves a matching task by: being dispatched by its solidarity cell to the task pool, during which the source address is its solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the agent's solidarity cell is configured to perform;"

Leong discloses this claim element when considered through the perspective and knowledge of a POSITA.[375] This element is substantially similar to claim element 1.8 and that discussion is incorporated-by-reference here.

#### 9. Claim Element 14.8: "searching the task pool for a task that is ready to be processed and has a function that the agent's solidarity cell can perform; and"

Leong discloses or renders obvious this claim element.[376] This element is substantially similar to claim element 1.9 and that discussion is incorporated-by-reference here.

---

[374] *Id.*, ¶214.

[375] *Id.*, ¶215.

[376] *Id.*, ¶216.

#### 10. Claim Element 14.9: "returning to its solidarity cell, during which the source address is the task pool's address, the destination address is the agent's solidarity cell's address, and the payload comprises a descriptor of the matching task."

Leong discloses this claim element.[377] This element is substantially similar to claim element 1.10 and that discussion is incorporated-by-reference here.

For all the reasons explained above, Leong renders obvious claim 14.[378]

## VI. GROUND 2: THE CHALLENGED CLAIMS ARE UNPATENTABLE OVER LEONG IN VIEW OF THE ETHERNET STANDARD.

As discussed in Section **Error! Reference source not found.**, the Challenged Claims are rendered obvious by Leong, when considered through the perspective and knowledge of a POSITA. Further, claims 1-14 are obvious over Leong in view of the Ethernet standard.[379]

The Ethernet standard describes the Ethernet networking standard which define aspects including the physical hardware and the Ethernet frame format.[380]

At the time of the '777 Patent's earliest priority date, Ethernet networks were well-known and commonplace. Leong describes how the processing units are

[377] *Id.*, ¶217.

[378] *Id.*, ¶218.

[379] *Id.*, ¶¶40, 219, 237.

[380] *Id.*, ¶¶50-54, 220.

coupled over an Ethernet network.[381] Leong doesn't, however, provide details regarding the Ethernet network.[382] The Ethernet standard provides such specifics, and a POSITA would have been motivated to refer to this specification for these details.[383]

Specifically, one may argue Leong lacks an express disclosure of the agent being a data frame comprising a source address, a destination address, and a payload, as set forth in claim element 1.7 (and claim element 14.6), and/or the specific values set in each of those data frame fields, as set forth in elements 1.8 and 1.10 (and claim elements 14.6, 14.7, 14.9).[384] But as evidenced by the prior art, this would have been obvious to a POSITA.[385]

The Ethernet standard shows a POSITA would have known how to structure a data frame comprising a source address, a destination address, and a payload.[386]

---

[381] Leong, 5:20-24; Weissman, ¶221.

[382] Weissman, ¶221.

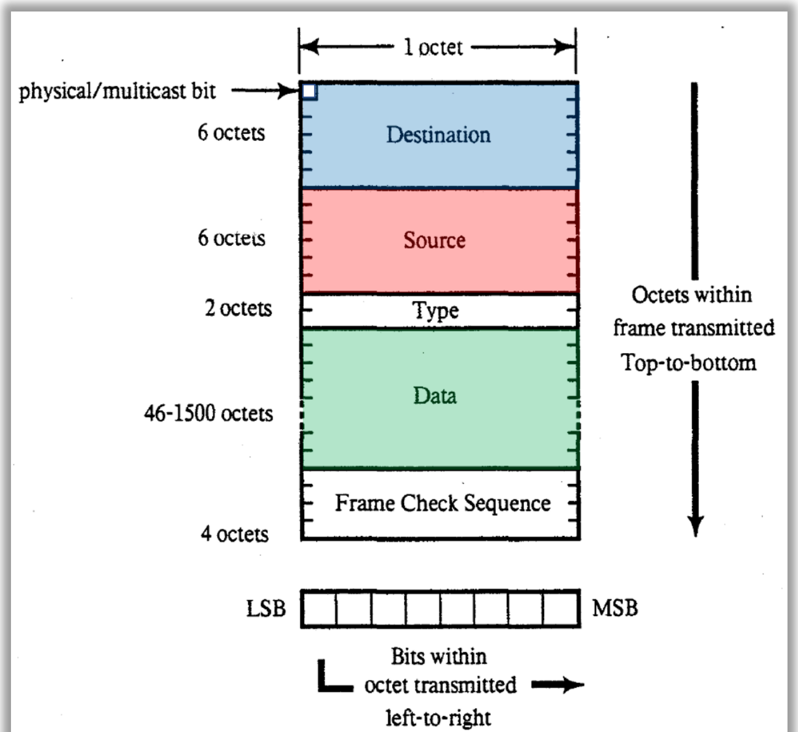
[383] *Id.*

[384] *Id.*, ¶222.

[385] *Id.*

[386] *Id.*, ¶223.

The Ethernet standard depicts the five fields in Figure 6-1, reproduced below, including source address (pink), destination address (light blue), and data (light green).[387]



"Ethernet" Figure 6-1 (Annotated)

The Ethernet standard describes the destination address field as specifying the device for which the frame is intended, the source address field as specifying the device sending the frame, and the data field containing the transmitted data.[388]

[387] Ethernet, 19-21; *see also* Kurose, 471-473, Figure 5.22; Weissman, ¶223.

[388] Ethernet, 19, 21-22; *see also* Kurose, 472-473 (describing the data, source address, and destination address); Weissman, ¶224.

It would have been obvious and a POSITA would have been motivated to apply the Ethernet standard's implementation details to Leong's processing units, which Leong already discloses use an Ethernet network.[389] The Ethernet standard provides specific teachings as to what is necessary to work in such a network.[390] And because Leong contemplates an Ethernet network, a POSITA would have had a reasonable expectation of success in applying the Ethernet standard's teachings.[391]

For all these reasons, the Challenged Claims would have been obvious to a POSITA over Leong in view of the Ethernet standard.[392]

## VII. GROUND 3: CLAIM 2 IS UNPATENTABLE OVER LEONG IN VIEW OF BATES AND/OR LEONG IN VIEW OF ETHERNET AND BATES

As discussed, the Challenged Claims, including claim 2, are rendered obvious by Leong (Section V), and/or rendered obvious by Leong in view of the Ethernet standard (Section VI). Further, claim 2 is obvious over Leong in view of Bates and/or Leong in view of the Ethernet standard and Bates.[393]

---

[389] Leong, 5:20-24; Weissman, ¶225.

[390] Weissman, ¶225.

[391] *Id.*, ¶225.

[392] *Id.*, ¶226.

[393] *Id.*, ¶¶40, 227, 237.

Bates describes a Cell processor with a main memory, one or more power processor units ("PPUs"), and multiple synergistic processing units ("SPUs").[394] Both the Bates PPU and SPUs can add tasks to a task queue.[395] And the Bates SPUs automatically obtain new tasks as needed.[396]

To the extent it is argued Leong's disclosures identified in Grounds 1 and 2 for claim 2 are insufficient, a POSITA would have found it obvious to provide for an additional type of task where the completion of tasks is monitored, as taught by Bates.[397]

Leong and Bates are in the same field of endeavor and contain overlapping disclosures with similar purposes.[398] Both Leong and Bates are directed to multi-processor systems with multiple processors for performing queued tasks.[399] Leong describes how each agent operates independently from each other and doesn't need any central coordination.[400] Although Bates describes how the PPU operates as a controller for the SPUs which handle the computational workload, a POSITA

---

[394] Bates, Abstract, ¶31; Weissman, ¶¶55-57, 228.

[395] Bates, ¶36; Weissman, ¶228.

[396] Bates, ¶42; Weissman, ¶228.

[397] Weissman, ¶229.

[398] *Id.*, ¶230.

[399] Leong, Abstract, 1:55-67; Bates, Abstract, ¶16; Weissman, ¶230.

[400] Leong, 2:40-48; Weissman, ¶230.

would nevertheless consider other aspects of Bates to be highly relevant, such as "task synchronization is useful when one task set must be completed before a subsequent task set can begin.[401]

It would have been obvious to combine Leong's system with Bates's specific teachings as to how tasks sequences can be executed in a multiprocessing system.[402] Leong omits specific details as to how different tasks signal their completion before other tasks are executed.[403] Bates provides those specific implementation details.[404] Instead of implementing a system involving reading/writing to memory (e.g., updating tasks in the bulletin board) and polling memory (e.g., checking the status of tasks in the bulletin board) to notify of completions, Bates teaches tasks signaling each other.[405] A POSITA would have sought Bates's technique because it teaches an efficient task completion system with implementation details.[406] Bates explains some SPU tasks depend on output data from a prior

---

[401] Bates, ¶¶32 (describing how the PPU operates as a controller), 53 (discussing task synchronization); Weissman, ¶230.

[402] Weissman, ¶231.

[403] *Id*.

[404] *Id*.

[405] *Id*.

[406] *Id*., ¶231.

task.[407] An SPU, upon task completion, can notify other SPUs through the use of a barrier or a task with an ID that is polled for completion.[408] Alternatively, Bates explains an interrupt is sent upon task completion.[409]

A POSITA would have been motivated to apply Bates's specific teachings regarding task synchronization to Leong's systems.[410] Leong itself contemplates how certain tasks can require the completion of other tasks before execution.[411] Leong does not, however, provide specific details as to how such a system can be implemented.[412] A POSITA would have been motivated to consider Bates's teachings because Bates also discloses a multiprocessor system and divides its computing jobs into tasks.[413]

A POSITA would have had a reasonable expectation of success.[414] Although Bates relies on a multiprocessor system with a PPU that controls processing,

---

407 Bates, ¶53; Weissman, ¶231.

408 Bates, ¶¶53-55, 59, FIG. 9B; Weissman, ¶231.

409 Bates, ¶59; Weissman, ¶231.

410 Weissman, ¶232.

411 Leong, 3:62-4:3, 7:46-52; Weissman, ¶232.

412 Weissman, ¶232.

413 Bates, ¶¶36-38; Weissman, ¶232.

414 Weissman, ¶233.

Bates's teachings regarding task synchronization are not so limited.[415] Bates teaches the SPU can notify the PPU of task completion through an ID that can be polled for completion, or through an interrupt sent to the PPU upon completion.[416]

Leong can be modified in view of Bates to implement those teachings.[417] For example, Leong's processing units already modify status information in the bulletin board to indicate task completion.[418] It would have been well within the skill of a POSITA to modify Leong's surveying units to poll tasks in the bulletin board, as taught by Bates.[419] Bates also teaches how a new task can monitor multiple tasks for completion.[420] For each of these modifications, Leong's bulletin board is notifying the surveying agent of task completion because Leong's status information, including that the task completion status is stored in the bulletin board.[421] A POSITA would have considered this a disclosure of the task pool notifying the CPU that a task has been completed.[422]

---

[415] *Id.*

[416] Bates, ¶59; Weissman, ¶233.

[417] Weissman, ¶234.

[418] Leong, 3:55-67, FIG. 7C (step 216); Weissman, ¶234.

[419] Weissman, ¶234.

[420] *Id.*

[421] *Id.*

[422] *Id.*

Accordingly, it would have been obvious to a POSITA to modify Leong in view of Bates's specific teachings regarding task synchronization.[423]

Dependent claim 2 is therefore obvious over Leong in view of Bates, or over Leong in view of the Ethernet standard, further in view of Bates.[424]

## VIII. DISCRETIONARY FACTORS FAVOR INSTITUTION.

Under §325(d), the Board shouldn't exercise discretionary denial. This Petition's prior art and obviousness arguments aren't similar to or cumulative of what the Office has considered. During prosecution, the Examiner never considered the teachings of Leong or Ethernet, nor the proposed obviousness combinations in Grounds 1-3.

Similarly, the six *Fintiv* factors confirm Petitioner's declaratory judgment district-court action provides no reason to deny institution under §314(a).

**Factor 1:** This is the first petition challenging the '777 Patent. Juniper filed a complaint for declaratory judgment of non-infringement in the Northern District of California ("CAND"). Juniper intends to request a stay in the district court if the

[423] *Id.*, ¶235.

[424] *Id.*, ¶236.

Board institutes trial. Recent statistics suggest district courts, in general, are becoming more likely to stay cases pending IPR.[425] And CAND grants 89% of motions to stay pending IPR.[426] Thus, it's likely a stay will be granted if the Board institutes IPR, which favors institution.

**Factor 2:** The district court matter is in its earliest stages. In the September 17, 2020 Joint Case Management Conference ("CMC") Statement, the parties proposed a trial date of March 7, 2022, but the court never entered that or any schedule so there is no trial date. And Patent Owner hasn't answered the complaint or made an infringement allegation, instead filing a motion to dismiss that it is still pending.[427] Once the district court rules on that motion, Petitioner anticipates that the court will enter a schedule.

The lack of a trial date "weighs significantly against" denying the Petition under §314(a).[428]

[425] Ex. 1015 (noting 70% success rate for stay motions pending IPR and rising).

[426] Ex. 1015, 1.

[427] Ex. 1016.

[428] *See Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (P.T.A.B. July 17, 2020) ("The fact that no trial date has been set weighs significantly against exercising our discretion to deny institution of the proceeding").

**Factor 3:** Investment in the district court action has been minimal and the parties have invested no time in the validity aspects of the case. Fact discovery hasn't opened, the parties haven't identified asserted claims, disputed claim terms, or exchanged infringement or invalidity contentions. This factor weighs against denying the Petition.[429]

**Factor 4:** This Petition challenges every claim. Given that Patent Owner has yet to identify its asserted claims, the early state of the district court proceeding makes it unclear if issues will overlap with the issues in this Petition.

**Factor 5:** The parties to Juniper's district-court action are the same, weighing in favor of discretionary denial.[430]

**Factor 6:** As fully detailed above, the interests of enhancing patent quality favor institution because the Petition's obviousness grounds are strong.[431]

In sum, the Board shouldn't exercise its discretion to deny institution under §§314(a) or 325(d).

---

[429] *See Sand Revolution II, LLC v. Cont'l Intermodal Group-Trucking LLC*, IPR2019-01393, Paper 24 at 10-11 (P.T.A.B. June 16, 2020).

[430] *Sand Revolution II,* Paper 24 at 12-13

[431] *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 14-15 (P.T.A.B. Mar. 20, 2020) (precedential) ("[I]f the merits of a ground raised in the petition seem particularly strong on the preliminary record, this fact has favored institution"); *Sand Revolution II,* Paper 24 at 13.

## IX. CONCLUSION

Leong confirms the claimed subject matter was not new or non-obvious. The Ethernet standard and Bates confirm certain implementation details that would have been obvious to a POSITA. Thus, the prior art discloses or renders obvious every limitation of the Challenged Claims.

In view of Grounds 1-3, there is a strong and reasonable likelihood the Challenged Claims are unpatentable. As such, the Board should institute IPR.

Respectfully submitted,

/Adam A. Allgood/
Adam A. Allgood
*Counsel for Petitioner*

## CERTIFICATE OF COMPLIANCE

Per 37 C.F.R. §42.24(a) and (d), the undersigned hereby certifies that the Petition complies with the type-volume limitation of 37 C.F.R. §42.24(a)(i) because, exclusive of exempted portions, it contains 13,939 words as counted by the word-processing program used to prepare it.

Dated: July 28, 2021

Respectfully submitted,

/Adam A. Allgood/

Adam A. Allgood (Reg. No. 67,306)
FISCH SIGLER LLP
5301 Wisconsin Avenue NW
Fourth Floor
Washington, DC 20015
Phone: 202.362.3536
Email: adam.allgood.ipr@fischllp.com

## CERTIFICATE OF SERVICE

The undersigned certifies service pursuant to 37 C.F.R. §§42.6(e) and 42.105(b) on the Patent Owner by Federal Express of a copy of this Corrected Petition for Inter Partes Review to the correspondence address of record for U.S. Patent No. 9,146,777:

Jennings, Strouss and Salmon, PLC
One East Washington Street, Suite 1900
Phoenix AZ 85004-2554

An electronic courtesy copy is also being served to litigation counsel at:

Allan Steyer
Steyer Lowenthal Boodrookas Alvarez & Smith LLP
235 Pine Street, 15th Floor
San Francisco, California 94104
asteyer@steyerlaw.com

Dated: July 28, 2021

Respectfully submitted,

/Adam A. Allgood/

Adam A. Allgood (Reg. No. 67,306)
FISCH SIGLER LLP
5301 Wisconsin Avenue NW
Fourth Floor
Washington, DC 20015
Phone: 202.362.3536
Email: adam.allgood.ipr@fischllp.com